# MAINVIEW® for DB2® Performance Reporter

## User Guide

Version 7.1

June 12, 2001

bmcsoftware

## Restricted Rights Legend

**BMC Software, Inc.**
2101 CityWest Blvd.
Houston TX  77042-2827
USA

## Contacting BMC Software

You can access the BMC Software Web site at **http://www.bmc.com**. From this Web site, you can obtain general information about the company, its products, special events, and career opportunities. For a complete list of all BMC Software offices and locations, go to **http://www.bmc.com/corporate/offices.html**.

### USA and Canada

**Address**
BMC Software, Inc.
2101 CityWest Blvd.
Houston TX  77042-2827

**Telephone**
713 918 8800  or
800 841 2031

**Fax**
713 918 8000

### Outside USA and Canada

| | |
|---|---|
| **Telephone** | (01) 713 918 8800 |
| **Fax** | (01) 713 918 8000 |

# Customer Support

You can obtain technical support by using Response Online™ (comprehensive information from the Web) or Response On Demand™. To expedite your inquiry, please see "Before Contacting BMC Software," below.

## Response Online

You can obtain technical support from BMC Software 24 hours a day, seven days a week by accessing the technical support Web site at **http://www.bmc.com/support.html**. From this site, you can

- read overviews about support services and programs that BMC Software offers
- find the most current information about BMC Software products
- search a database for problems similar to yours and possible solutions
- order or download product documentation
- report a problem or ask a question
- subscribe to receive e-mail notices when new product versions are released
- find worldwide BMC Software support center locations and contact information, including e-mail addresses, fax numbers, and telephone numbers

## Response On Demand

In the USA and Canada, if you need technical support and do not have access to the Web, call 800 538 1872. Outside the USA and Canada, please contact your local support center or your local sales office for assistance.

## Before Contacting BMC Software

Before you contact BMC Software, have the following information available so that a technical support analyst can begin working on your problem immediately:

- product information
  - product name
  - product version (release number)
  - license number and password (trial or permanent)
- operating-system and environment information
  - machine type
  - operating system type, version, and service pack or program temporary fix (PTF)
  - system hardware configuration
  - serial numbers
  - related software (database, application, and communication) including type, version, and service pack or PTF
- sequence of events leading to the problem
- commands and options that you used
- messages received (and the time and date that you received them)
  - product error messages
  - messages from the operating system, such as `file system full`
  - messages from related software

# Contents

# Figures

# Tables

# About This Book

This book describes how to use MAINVIEW for DB2 Performance Reporter. It can be used by

- DB2 system and database administrators who capture and review the activity of each day's DB2 session and track system and application profiles over time
- System programmers who isolate recurring problems in the DB2 environment
- Data center managers who make hardware and resource acquisition decisions based on long-term trends

Before using this book, you must be familiar with IBM's MVS batch job execution and the IBM DATABASE 2 program product.

**Note:** Throughout this manual, Performance Reporter refers to MAINVIEW for DB2 Performance Reporter and MVDB2 refers to MAINVIEW for DB2.

The DMR acronym for the product is used occasionally in this book and in many online panels and messages.

## How This Book Is Organized

This *Performance Reporter User Guide* is organized as follows:

- Part 1, "Using Performance Reporter" on page 1 is an overview of how to use Performance Reporter. It includes:
  - The SMF extract and table update procedure (DPRSMF)
  - The data summary and purge process (DPSUMLD)
  - The data reporting facilities that produce batch reports and charts
- Part 2, "Reports from SMF" on page 89 describes each of the reports that can be produced directly from SMF data.
- Part 3, "Reports from DB2 Tables" on page 333 describes each of the reports (DPRREPT and QMF) produced by Performance Reporter from DB2 tables in detail.
- Part 4, "Performance Data Tables" on page 443 describes the Performance Reporter performance data tables.
- Part 5, "Appendixes" on page 549 contains instructions on receiving help from Customer Support. It also contains tables describing the BBSAMP and BBPARM data set members and explanations on how to use the MAINVIEW product libraries and collect SMF data.
- Part 6, "Index" on page 557 contains the Index.

## Conventions Used in This Book

The following syntax notation is used in this manual. Do not type the special characters.

- Brackets [ ] enclose optional parameters or keywords.
- Braces { } enclose a list of parameters; one must be chosen.
- A vertical line | separates alternative options; one can be chosen.
- An underlined parameter is the default.
- AN ITEM IN CAPITAL LETTERS indicates exact characters; usage can be all uppercase or lowercase.
- Items in lowercase letters are values you supply.

## Recommended Reading

The following books are referenced in this edition:

- *DB2 Administration Guide* by IBM
- *Query Management Facility Learner’s Guide* by IBM
- *Using MAINVIEW®*
- *Quick Start with MAINVIEW®*
- *OS/390 and z/OS Installer Guide*
- *MAINVIEW® Installation Requirements Guide*
- *MAINVIEW® Common Customization Guide*
- *MAINVIEW® Alternate Access Implementation and User Guide*
- *Implementing Security for MAINVIEW® Products*
- *MAINVIEW® Administration Guide*
- *MAINVIEW® for DB2® Release Notes*
- *Getting Started with MAINVIEW® for DB2® and RxD2™*
- *MAINVIEW® for DB2® User Guide Volume 1: Views*
- *MAINVIEW® for DB2® User Guide Volume 2: Analyzers/Monitors*
- *MAINVIEW® for DB2® User Guide Volume 3: Traces*
- *MAINVIEW® for DB2® Performance Reporter User Guide*
- *RxD2™ User Guide*
- *MAINVIEW® for DB2® Customization Guide*

## Related Reading

This book is included as part of the MAINVIEW library, which documents all your MAINVIEW products and the tasks associated with using these products.

## MAINVIEW Library

The MAINVIEW library documents these products:

- CMF® MONITOR Online (CMF)
- IMSPlex System Manager™ (IPSM)
- MAINVIEW® Alarm Manager (MVALARM)
- MAINVIEW® AutoOPERATOR™ (AO)
- MAINVIEW® Explorer
- MAINVIEW® FOCAL POINT™
- MAINVIEW® for CICS (MVCICS)
- MAINVIEW® for DB2® (MVDB2)
- MAINVIEW® for DBCTL (MVDBC)
- MAINVIEW® for IMS (MVIMS)
- MAINVIEW® for IP (MVIP)
- MAINVIEW® for MQSeries (MVMQS)
- MAINVIEW® for OS/390 (MVMVS)
- MAINVIEW® for UNIX System Services (MVUSS)
- MAINVIEW® for VTAM (MVVTAM)
- MAINVIEW® for Websphere
- MAINVIEW® VistaPoint™ (MVVP)
- Plex Manager (PLEXMGR)

The MAINVIEW library is organized into these three categories:

- Installer documentation
- Administrator documentation
- User documentation

Each book within these categories contains information about specific types of tasks. The following figure shows how this book relates to the other books in the MAINVIEW library.

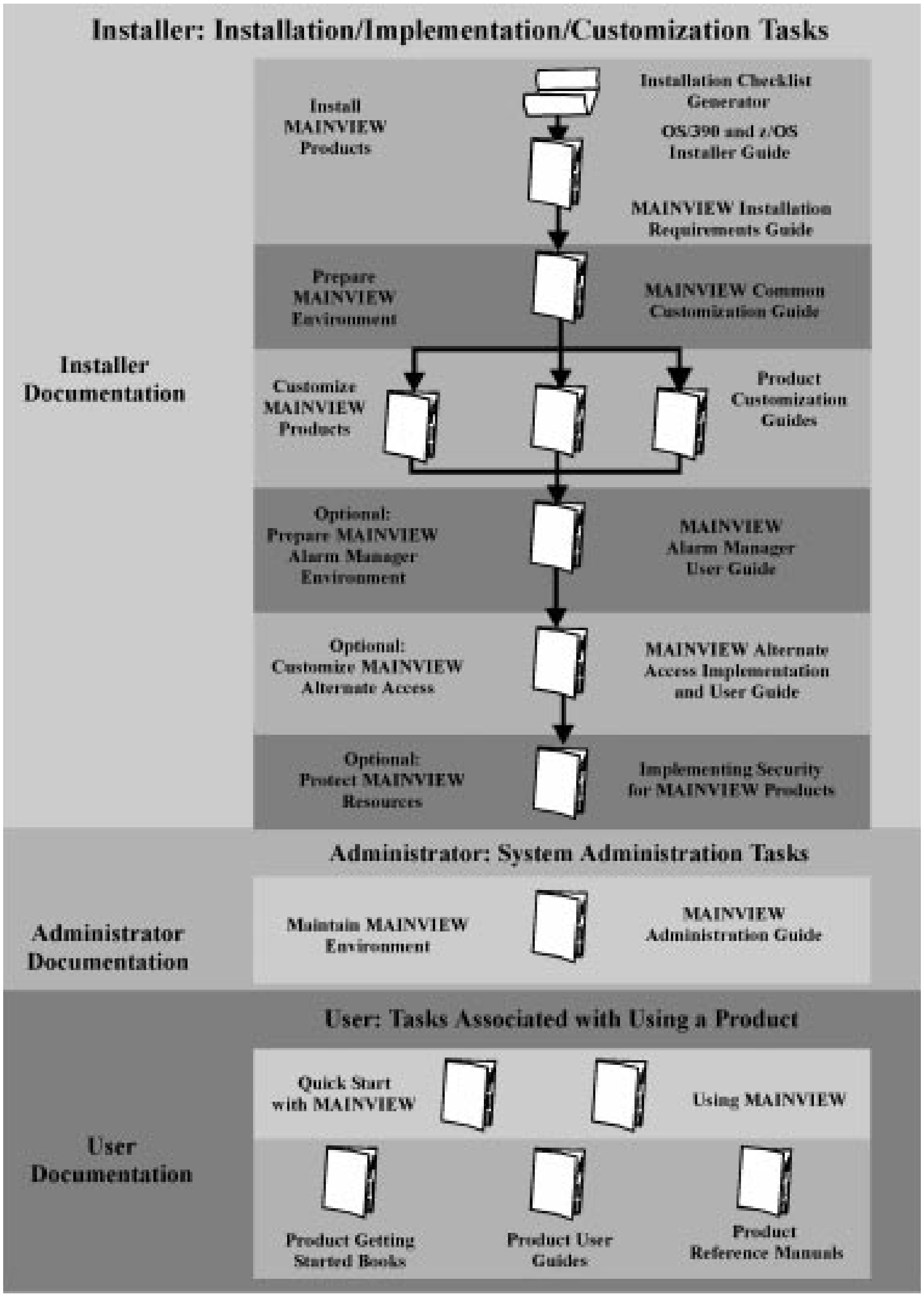

Installer: Installation/Implementation/Customization Tasks
Installer Documentation
Install MAINVIEW Products
Installation Checklist Generator
OS/390 and z/OS Installer Guide
MAINVIEW Installation Requirements Guide
Prepare MAINVIEW Environment
MAINVIEW Common Customization Guide
Customize MAINVIEW Products
Product Customization Guides
Optional: Prepare MAINVIEW Alarm Manager Environment
MAINVIEW Alarm Manager User Guide
Optional: Customize MAINVIEW Alternate Access
MAINVIEW Alternate Access Implementation and User Guide
Optional: Protect MAINVIEW Resources
Implementing Security for MAINVIEW Products
Administrator: System Administration Tasks
Administrator Documentation
Maintain MAINVIEW Environment
MAINVIEW Administration Guide
User: Tasks Associated with Using a Product
User Documentation
Quick Start with MAINVIEW
Using MAINVIEW
Product Getting Started Books
Product User Guides
Product Reference Manuals

Figure 1. Organization of MAINVIEW Documentation

The following books in the MAINVIEW library are used with all MAINVIEW products:

- *Quick Start with MAINVIEW* gives a brief overview to help you quickly get started using all your MAINVIEW products.
- *Using MAINVIEW* gives a more detailed description of how to use your MAINVIEW products. If you have more than one MAINVIEW product, this book will help you understand how all your MAINVIEW products work. This book will also help you use your products together and take advantage of the integration of all MAINVIEW products.
- The *OS/390 and z/OS Installer Guide* and *MAINVIEW Installation Requirements Guide* give instructions for basic installation of the product libraries.
- The *MAINVIEW Common Customization Guide* and the *MAINVIEW Administration Guide* provide customization and administration instructions for all MAINVIEW products.
- The *MAINVIEW Alternate Access Implementation and User Guide* describes how to use the MAINVIEW Alternate Access component. This component provides EXCP and VTAM communication to BMC Software products through ISPF without requiring a TSO subsystem to be active.
- The *MAINVIEW Alarm Manager User Guide* describes how to generate alarms when thresholds from MAINVIEW product views are exceeded.
- *Implementing Security for MAINVIEW Products* describes how to implement security for MAINVIEW for DB2 with the external security manager installed at your site.
- The *MAINVIEW Explorer Implementation and User Guide* explains how to install and use MAINVIEW Explorer, which provides access to MAINVIEW products from a Web browser running on a Windows workstation.

## MAINVIEW for DB2 Library

The following documentation gives specific information about the MAINVIEW for DB2 and RxD2 products:

- *MAINVIEW for DB2 Release Notes* summarize the new features in this release of MAINVIEW for DB2 and RxD2. These notes enable you to quickly see what is new.
- The *MAINVIEW for DB2 Master Index* includes index entries for all the product-specific books in the MAINVIEW for DB2 library. This index shows where you can find information for specific topics in individual MAINVIEW for DB2 and RxD2 manuals.
- *Getting Started with MAINVIEW for DB2 and RxD2* is an introduction for new users of these products. The book helps you use these products to solve problems more effectively in a short time.
- The *MAINVIEW for DB2 User Guide* (Volumes 1, 2, and 3) describes how to use the online views, analyzer, monitor, and trace services for the DB2 database administrator, system programmer, or performance analyst.
- The *MAINVIEW for DB2 Performance Reporter User Guide* describes how to create statistical batch reports about application activity and DB2 performance and resource usage for the database administrator, system programmer, or performance analyst.

- The *RxD2 User Guide* describes how to install and use the RxD2 product, which provides access to DB2 from REXX. This product also provides tools to query the DB2 catalog, issue dynamic SQL, test DB2 applications, analyze EXPLAIN data, generate DDL or DB2 utility JCL, edit DB2 table spaces, perform security administration, and much more. MAINVIEW for DB2 contains numerous hyperlinks to RxD2.

- The *MAINVIEW for DB2 Customization Guide* provides product-specific tailoring instructions. Use this book in conjunction with the *MAINVIEW Installation Requirements Guide* and the *MAINVIEW Common Customization Guide* during the customization process.

- Online tutorials are available by selecting option T from the MAINVIEW for DB2 Primary Option Menu or by pressing HELP (PF1/13) from the product application panels.

# Part 1. Using Performance Reporter

This part describes how to use Performance Reporter.

**Note:** To produce batch reports, you must first customize the MAINVIEW for DB2 Performance Reporter component.

BMC Software provides AutoCustomization procedures, described in the *OS/390 and z/OS Installer Guide* so that you can tailor your product automatically. The “Customizing Performance Reporter” chapter in the *MAINVIEW for DB2 Customization Guide* describes the steps for tailoring your product manually.

# Chapter 1. Introduction

Performance Reporter is an offline analysis system that produces reports and plots that can be used to evaluate DB2 system and application performance. These evaluations can be used for DB2 planning, forecasting, and performance management.

The data is captured from the trace records written by DB2 to SMF. Performance Reporter processes accounting, statistics and audit records. They are loaded into DB2 tables where the power of Structured Query Language (SQL) and Query Management Facility (QMF) can be used to fulfill your reporting needs. Queries either distributed with the product or modified or developed on-site can be used to generate reports. Data summarization and purge routines can be used to control the amount of data while still allowing for both short- and long-term historical reporting.

Several reports can be produced from SMF extract files without loading the data into DB2 tables. This includes accounting short and long reports and statistics short and long reports.

**Note:** Reports on detail performance trace records (SQL and so on) are processed as part of the Application Trace batch trace print facility, from trace logs or in some cases from SMF data. See “Printing a Trace” in Volume 2 of the *MAINVIEW for DB2 User Guide*.

Figure 2 summarizes the processing flow used to produce these reports and charts with Performance Reporter.

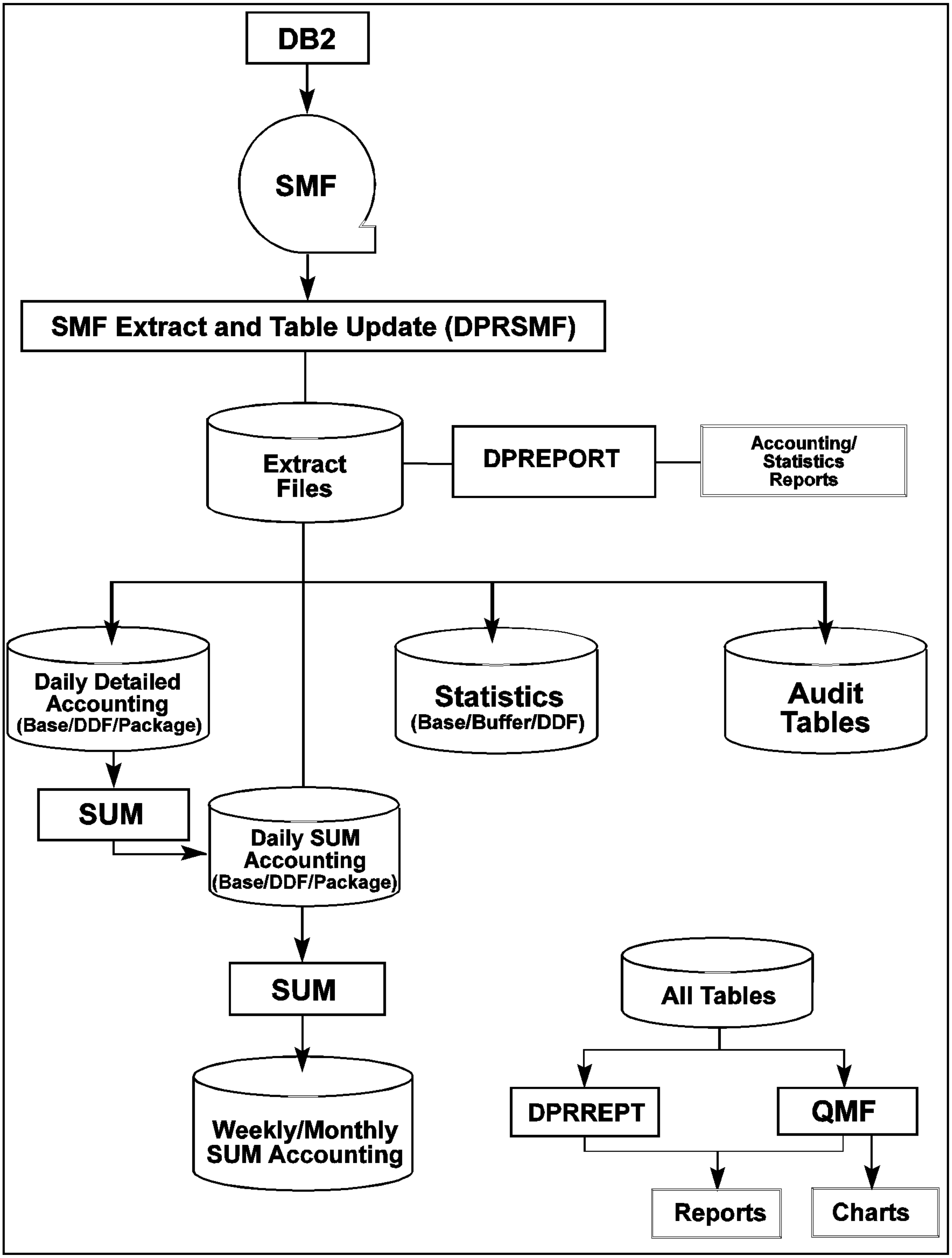

DB2
SMF
SMF Extract and Table Update (DPRSMF)
Extract Files
DPREPORT
Accounting/ Statistics Reports
Daily Detailed Accounting (Base/DDF/Package)
Statistics (Base/Buffer/DDF)
Audit Tables
SUM
Daily SUM Accounting (Base/DDF/Package)
SUM
Weekly/Monthly SUM Accounting
All Tables
DPRREPT
QMF
Reports
Charts

Figure 2. Performance Reporter System Overview

## Performance Reporter Input Data

Performance Reporter processes DB2 trace records written to SMF, including the DB2 statistics, accounting, and audit SMF records. DB2 captures data on its activity as it occurs and writes this information to the SMF data sets as statistics, accounting, audit, and performance trace records. The number of records produced and the amount of data collected and stored in these records is determined by the DB2 trace options in effect within the DB2 subsystem. For a short description of these DB2 trace options, see Appendix C, “SMF Data Collection” on page 555. For more information on these options and the resulting differences in DB2 trace records, see the *IBM DATABASE 2 Administration Guide*.

## SMF Data Extraction and Reporting

The SMF data is processed by the MVDB2 batch job DPRSMF, which extracts the DB2 statistics, accounting, and audit SMF records and stores the data in the detailed performance data tables.

The accounting and statistics extract files created in DPRSMF can also be used to create reports without loading the data into DB2 tables. These reports can be produced either in the DPRSMF job or the extract files can be kept for later processing in the DPRSMFR job.

## Performance Data Tables in DB2

The performance data tables are the main source of historical information for Performance Reporter. These tables are created initially as detailed accounting, statistics, buffer, and audit tables. There is

- One row for each SMF accounting record
- One row for each PACKAGE in accounting with class 7/8 active
- One row for each DDF destination in accounting records (optional)
- One row for each SMF statistics interval generated by a DB2 system
- One row for each SMF statistics interval with buffer pool data summarized for 4K and 32K pools
- One row for each SMF statistics interval per single buffer pool (optional)
- One row for each DDF destination per statistics interval (optional)
- One row for each SMF audit record (optional)

Accounting records, which may occur in high volumes, can be loaded from SMF into a DB2 detail table or summarized first and loaded directly into a summary table.

## Data Summarization and Purge

The data in DB2 detail accounting tables or package accounting table can be extracted, reduced, and stored in the DB2 summary accounting tables by a summarization process. The amount of data in the detail statistics and accounting tables can be regulated by a time-controlled purge process. Performance Reporter reports can be created from either the detailed or summary tables.

In Performance Reporter, the DPSUMLD program reads the DB2 detail accounting table, summarizes the input data, and stores the data in the DB2 summary accounting table. Information in DB2 detail accounting tables can be summarized for a time period, such as one day. Abstracted data is grouped in plan execution rows, which summarize information for each plan executed. You can specify the time interval to be included in a summary period and which accounting key values are used to determine uniqueness for a summary record. The same process is done for the package accounting table with additional summary keys provided.

## Reports from the DB2 Tables

Reporting from the performance data tables within Performance Reporter can be performed through QMF or the supplied reporting program DPRREPT, which executes SQL queries.You can alter the supplied queries or create others for specific reporting requirements, or use any in-house DB2 query tool.

Table 1 lists by default name each of the tables that can be created in Performance Reporter. The related BBPARM and BBSAMP members are also shown.

Table 1. List of Performance Reporter Tables

| Default Table Name Description | Default Table Space Name | Create Table Member (UBBSAMP) | Report Members (BBPARM) | QMF Members (BBSAMP) |
|---|---|---|---|---|
| DMRPR.DMRACDTL (detail accounting table) | DMRPRTAD | DPCTACDT | ACxxxxxx | QACxxxxx |
| DMRPR.DMRACSUM (summary accounting table) | DMRPRTAS | DPCTACSM | SAxxxxxx | QSAxxxxx |
| DMRPR.DMRACSM2 (summary-2 accounting table) | DMRPRTA2 | DPCTACS2 | SAxxxxxx | QSAxxxxx |
| DMRPR.DMRABDTL (detail buffer accounting table) | DMRPRTID | DPCTABDT | ACxxxxxx | QACxxxxx |
| DMRPR.DMRABSUM (summary buffer accounting table) | DMRPRTIS | DPCTABSM | SAxxxxxx | QSAxxxxx |
| DMRPR.DMRABSM2 (summary-2 buffer accounting table) | DMRPRTI2 | DPCTABS2 | SAxxxxxx | QSAxxxxx |

Table 1. List of Performance Reporter Tables (Continued)

| Default Table Name Description | Default Table Space Name | Create Table Member (UBBSAMP) | Report Members (BBPARM) | QMF Members (BBSAMP) |
|---|---|---|---|---|
| DMRPR.DMRADDTL (detail DDF accounting table) | DMRPRTDD | DPCTADDT | ACxxxxx | QACCPDFx<br>QACCPFFA<br>QACDDFxx<br>QACOVDFx |
| DMRPR.DMRADSUM (DDF summary accounting table) | DMRPRTDS | DPCTADSM | SAxxxxxx | QSACPDFx<br>QSADDFLx<br>QSADDFRx<br>QSAOVDFx |
| DMRPR.DMRADSM2 (DDF summary-2 accounting table) | DMRPRTD2 | DPCTADS2 | SAxxxxxx | QSACPDFx<br>QSADDFLx<br>QSADDFRx<br>QSAOVDFx |
| DMRPR.DMRAPDTL (detail package accounting table) | DMRPRTPD | DPCTAPDT | ACxxxx | QACxxxx |
| DMRPR.DMRAPSUM (package summary accounting table) | DMRPRTPS | DPCTAPSM | SAxxxx | QSAxxxx |
| DMRPR.DMRAPSM2 (package summary-2 accounting table) | DMRPRTP2 | DPCTAPS2 | SAxxxx | QSAxxxx |
| DMRPR.DMRSTAT (detail statistics table) | DMRPRTSS | DPCTST | STxxxxxx | QSTxxxxx<br>QSTOVRxx |
| DMRPR.DMRSBFDT (detail buffer statistics table) | DMRPRTBD | DPCTBFDT | STxxxxx | QSTxxxx |
| DMRPR.DMRSTDF (DDF statistics table) | DMRPRTSF | DPCTSD | STOVDFT<br>STOVRxx | QSTOVDFT<br>QSTOVRxx |
| DMRPR.DMRAUSUM (audit summary table) | DMRPRAUS | DPCTAUSM | AUSUM<br>AUDTL | QAUSUM<br>QAUDTL |
| DMRPR.DMRAUFAL (authorization failures table) | DMRPRAUF | DPCTAUFL | AUFAIL | QAUFAIL |
| DMRPR.DMRAUGRV (authorization control - GRANTs / REVOKEs table) | DMRPRAUG | DPCTAUGR | AUDGRV | QAUDGRV |
| DMRPR.DMRAUDDL (DDL access table) | DMRPRAUD | DPCTAUDL | AUDDL | QAUDDL |
| DMRPR.DMRAUDML (DML access table) | DMRPRAUM | DPCTAUDM | AUDML | QAUDML |
| DMRPR.DMRAUDMB (DML at BIND table) | DMRPRAUB | DPCTAUDB | AUDMLB | QAUDMLB |

Table 1. List of Performance Reporter Tables (Continued)

| **Default Table Name Description** | **Default Table Space Name** | **Create Table Member (UBBSAMP)** | **Report Members (BBPARM)** | **QMF Members (BBSAMP)** |
|---|---|---|---|---|
| DMRPR.DMRAUCHG (authorization ID change table) | DMRPRAUC | DPCTAUCH | AUCHNG | QAUCHNG |
| DMRPR.DMRAUUTL (utility access table) | DMRPRAUU | DPCTAUUT | AUUTIL | QAUUTIL |

## Table Release Dependencies and Migration

The table contents change from one release of MVDB2 to another, mainly to add columns supporting new DB2 release information. When migrating to a new release of MVDB2, you must define the new tables with unique names. All related jobs and reports must be run only with the tables created by that release of MVDB2.

When you are ready to move a new release into production use, you may want to migrate the data in the previous tables to the new table format. The member DPJMIGR in BBSAMP contains a sample job to migrate tables from the previous release. See the *MAINVIEW for DB2 Customization Guide* for more information.

# Chapter 2. SMF Extract, Reporting, and Table Update (DPRSMF)

This chapter describes the DPRSMF batch job. The purpose of this procedure is to extract DB2 SMF statistics, accounting, and audit records and load this data into the performance data tables in Performance Reporter.

The DPRSMF job stream can be tailored to produce accounting and statistics reports during the daily SMF extract, either in addition to, or in place of, loading the data into DB2 tables. The extract files can also be saved for later reporting with the DPRSMFR job.

The SMF data set and control statements describing the DB2 tables to be loaded are the input to this procedure. The output of this process is the extracted data added to the detail performance data tables and, optionally, accounting and statistics reports or extract files.

## DPRSMF Job Control Statements

SMF data extraction is a six-step batch execution procedure (see Figure 3 on page 20). This procedure can be run stand-alone against the SMF data sets or can be incorporated into an existing job that also processes this data. It can be broken down into its component parts. Generation data groups or other permanent data set types can be used to store and pass the data between runs of the various steps. Some steps, such as the table load, can be delayed until end of the day processing.

The procedure symbolics can be specified to disable either the summarization / load steps or the report steps. The accounting and statistics extract files can be temporary or kept for additional reporting later. (See Chapter 4, “SMF Reporting Facilities (DPREPORT)” on page 57.)

The six SMF data extraction steps are as follows:

1. Extract the DB2 statistics, accounting, and audit records from the input SMF file, creating temporary statistics, accounting, and audit files.

   Control statements allow the selection of data of only certain specified record types or from specific DB2 subsystems.

2. Create discrete interval records from the cumulative statistics records. To consolidate any statistics intervals that span an SMF tape, this step maintains a spin file which resolves issues caused by processing the SMF tape out of sequence.

3. Build DB2 load control statements, summarize accounting and DDF accounting records, and purge old records.

   Control statements allow the specification of the tables in which the extracted data is to be loaded and whether summarization should occur before the load. Other statements allow existing table data to be purged or unloaded.

4. Load the performance data tables. You must have the proper DB2 authorization for this step.

5. Print a statistics report. The default is a statistics long report for the total interval included in the input data.

   The control statements can be modified to select data by date, time, relative day, SMFID, or DB2 ID. A different summarization interval can also be specified.

6. Print an accounting report. The default is an accounting long report by plan for the total interval included in the input data.

   The control statements can be modified to select data by date, time, relative day, SMFID, DB2 ID, or various thread identifiers. The summarization keys and interval can also be modified.

The following list describes the job control language (JCL) statements used in Figure 3 on page 20:

## SMF Extract and File Creation

STEP 1

Specifies the name of the Performance Reporter statistics, accounting, and audit file creation program as

```
PGM=DPRDSMF
```

and the region required to run the program.

STEPLIB DD

Defines the program library containing the DPRDSMF load module.

DPRLOG DD

Defines the data set for the program results summary, which indicates input and output record counts.

SYSOUT DD

Defines the output class for sort messages.

SYSUDUMP DD

Defines the dump data set for problem determination.

DPDACCT DD

Contains the accounting records created from the DB2 SMF 101 accounting records. This file is passed to Step 3 ACCTDD and to Step 6 DPDACCT.

DPDSTAT DD

Contains the statistics records created from the DB2 SMF 100 statistics records. This file is passed to Step 2.

DPDAUDIT DD

Contains the audit records created from the DB2 SMF 102 audit records. This file is passed to Step 4.

SORTIN DD

Defines the data (SMF data set) to be extracted and sorted.

SORTWKnn DD

Defines work data sets for data sorting; nn is a numeric.

DPRCNTL DD

The following control statements are valid:

SSID=

Defines the DB2 subsystem IDs desired when you wish to select only certain DB2 subsystem IDs from the SMF data as it is read. If this operand is omitted, all subsystems are selected. As many as 10 entries may be entered, enclosed in parentheses and separated by commas.

SHORTREC=(YES|NO)

Allows selection by DB2 subsystem ID from SMF data with records shorter than 18 bytes. The default is NO.

**Note:** This control statement is recommended only if you have specified SSID(s) and receive message DPR0104I or SYNCSORT message WER250A. You will receive this message if some of the SMF records are shorter than 18 bytes and the VLSHRT option is not applicable to your SORT routine.

If you are using DFSORT Release 11.1 or higher, see DFSPARM DD on page 13 for more information.

TYPE=

Allows selection of only certain types of SMF accounting and statistics data. The possible types are

| | |
|---|---|
| ACCT | Accounting base data |
| PKGACCT | Package accounting data |
| DDFACCT | DDF accounting data |
| BUFACCT | Detail buffer accounting data per pool |
| STAT | Statistics data, including buffer statistics |
| DDFSTAT | DDF statistics data |
| BUFSTAT | Detail buffer statistics data per pool |

These keywords may be entered in any combination, enclosed in parentheses and separated by commas.

If this entry and TYPE2 are not coded, all accounting and statistics types are selected.

TYPE2=

Allows selection of audit data types. The possible types are

| | |
|---|---|
| AUDFAIL | Authorization failures data |
| AUDGRV | Authorization control - GRANTs / REVOKEs data |
| AUDDDL | DDL access data |
| AUDDML | DML access data |
| AUDDMB | DML at BIND |
| AUDCHG | Authorization ID change data |
| AUDUTL | Utility access data |
| AUDSUM | Summary data |

These keywords may be typed in any combination, enclosed in parentheses and separated by commas.

Audit data types are not selected unless TYPE2 is specified.

EXCLUDEPKG=

Allows specific package accounting records to be excluded.

Possible keywords are

| | |
|---|---|
| PROGRAMNAME | Program name (package ID or DBRM) |
| FIRSTPKG | First package or DBRM executed |
| CORRID | Correlation ID value |
| CONNECTION | Connection name |
| CONNTYPE | Connection type |
| PLANNAME | Plan name |
| AUTHID | Primary authorization ID |

Up to eight values can be specified for each keyword. For example:

```
EXCLUDEPKG PROGRAMNAME=DSNESM68,
FIRSTPKG=(PKG11,PKG12,PKG13XXXXXXXXXXXXX,TSMCOM2,
         PKG15,PKG16,PKG17XXXXXXXXXXXXX,TSMCOMX),
CORRID=(PKGCORR1,PKGCORR2,PKGCORR3XXXX,BOLHHH4),
CONNECTION=(PKGCONN1,PKGCONN2,PKGCONN3,DB2CALL),
CONNTYPE=(PKGCONT1,PKGCONT2,PKGCONT3,'DB2 CALL'),
PLANNAME=(PKGPLAN1,PKGPLAN2,PKGPLAN3,DSNTIA51),
AUTHID=(PKGAUTH1,PKGAUTH2,PKGAUTH3,BOLCJN2)
```

excludes from package accounting (PKGACCT) all records with values matching those specified.

DATE=

Can be used to select only records from specific date ranges, as follows:

```
DATE=01JAN2001-01FEB2001 Records for all of JAN 2001
DATE=*_1-*               Records from yesterday
DATE=*_3                 Records for previous 3 days plus today
```

TIME=

Can be used to select only records from specific time ranges, as follows:

```
TIME=0800-1700           First shift records only
TIME=0800                Records from 8AM to midnight
```

**Timezone Adjustment Process**

The next three keywords refer to the timezone adjustment process.

You should only supply the parameters in TZPARM when local time is not equal to Greenwich mean time at your shop (a non-zero value is in SYS1.PARMLIB member CLOCKXX). Type 100 and 101 SMF records for DB2 contain Greenwich mean time. TZPARM must contain the correct parameters for the DPRSMF program to convert Greenwich mean time to local time.

SMFID=

Defines the SMF ID for which the following timezone parameters apply. If the timezone parameters apply for all SMF IDs which may be processed, it is not necessary to include SMFID.

It is invalid to include some TZSTART/TIMEZONE combinations without a qualifying SMFID, and then other TZSTART/TIMEZONE combinations after the qualifier.

TZSTART=

Defines the starting time after which the adjustment is made to any datetime fields. Its data field is in datetime format.

TIMEZONE=

Optional. Defines the adjustment amount.

For example, sample parameters for the first SMF group are

```
SMFID=SYSA
TZSTART=1991-04-01-02.00.00,TIMEZONE=W.05.00.00
TZSTART=1991-10-28-02.00.00,TIMEZONE=W.06.00.00
```

Sample parameters for the second SMF group are

```
SMFID=SYSB
TZSTART=1991-04-01-01.00.00,TIMEZONE=W.05.00.00
TZSTART=1991-10-28-04.00.00,TIMEZONE=W.06.00.00
```

Syntax rules are as follows:

- No blank spaces are allowed between parameters within a line.
- All preceeding zeros are required for month, day, and time. For example, specify 04 for the month of April.
- A maximum of 40 TZSTART/TIMEZONE entries are allowed.
- Because of the difference between daylight savings and standard timezones, a minimum of two datetime entries is required for each SMF group entry.
- If the SMFID is not specified, the datetime entries apply to all DB2 SMF records encountered.
- When the SMFID is specified, the associated datetime entries apply only to the DB2 SMF records with the specified SMFID.
- The timestamp of the datetime entry indicates the local time when the change of timezone occurs. For example:

  ```
  TZSTART=1991-04-01-02.00.00,TIMEZONE=W.05.00.00
  ```

  indicates the TIMEZONE value was changed to W.05.00.00 at local time 1991-04-01-02.00.00.

DFSPARM DD

Specifies the following override option (for DFSORT users only):

VLSHRT

Temporarily overrides the VLSHRT installation option. DFSORT continues processing if a variable-length input record is too short to contain all specified INCLUDE compare fields.

**Note:** This override parameter is necessary only if you receive the following messages in the job output:

- Message DPR0104I
- Return code 16
- An error message from the sort routine indicating that an SMF record is too short

This method of processing short SMF records provides better performance than using the SHORTREC control statement. However, this option works only with DFSORT Release 11.1 or higher. If using any other sort routine, you need to add the SHORTREC control statement.

## General Syntax Rules

The general rules for coding the operands are the same as those for the operands in SUM/PURGE.

- If operands are to be continued, end the first line with a comma-blank.
- Comments are allowed after a blank.
- Comment lines are designated by an asterisk in column 1.
- If a value contains special characters such as blank, equal sign, or comma, enclose the value in quotation marks.
- Multiple values for a keyword are enclosed in parentheses and separated by commas.
- Multiple keywords may be coded on a single line, separated by commas.
- Undefined keywords, invalid values, missing delimiters, and such generate error messages.

## Statistics Consolidation

STEP 2
Specifies the name of the Performance Reporter statistics consolidation program as

PGM=DPRSTAT

and the region required to run the program.

STEPLIB DD
Defines the program library containing the DPRSTAT load module.

DPRLOG DD
Defines the data set for DPRSTAT results.

SYSOUT DD
Defines the output class.

SYSUDUMP DD
Defines the dump data set for problem determination.

SORTIN DD
Input file. Defines the statistics record file passed from Step 1.

DPDSPIN DD
Defines a spin file to retain SMF statistics records that end a period to resolve intervals on subsequent tape processing.

DPDSTAT DD
Output file. Defines the final statistics record file ready to be loaded to DB2 tables. This file is passed to Step 4 STATDD and to Step 5 DPDSTAT.

SORTWKnn DD
Defines work data sets for data sorting; nn is a numeric.

STEP 3
Specifies the name of the Performance Reporter summary load control program as

```
PGM=DPSUMLD
```

**Note:** The procedure symbolic SUMPGM can be changed to IEFBR14 to disable this step.

STEPLIB DD
Defines the program library containing the DB2 load library and the Performance Reporter load library.

DPSYSOUT DD
Defines the data set for DPSUMLD results.

SYSOUT DD
Defines the output class.

SYSUDUMP DD
Defines the dump data set for problem determination.

SORTWKnn DD
Defines work data sets for data sorting; nn is a numeric.

ACCTDD DD
Defines the input data set for the ACCT SUMMARIZE process.

SUMACCT DD
Defines the output data set for the ACCT SUMMARIZE process.

DDFSACCT DD
Defines the output data set for the DDFACCT SUMMARIZE process.

PKGSACCT DD
Defines the output data set for the package PKGACCT SUMMARIZE process.

BUFSACCT DD
Defines the output data set for the Buffer Pool BUFACCT SUMMARIZE process

DPLOAD DD
Defines the data set for DB2 LOAD control statements.

DPSYSIN DD
Defines the data set containing the control statements for the DPSUMLD program.

Refer to “DPSUMLD Control Statements” on pages 27 through 35 for more information on the control statements.

## Load into DB2 Tables

STEP 4
Specifies the name of the DB2 Utility program as

PGM=DSNUTILB

and the region required to run the program.

Steps to load DB2 tables use the DB2 Load Utility, DSNUTILB. You must modify your JCL if your installation uses a different utility. The utility must be able to interpret LOAD control statements using the same format that DSNUTILB uses.

**Note:** The procedure symbolic LOADPGM can be changed to IEFBR14 to disable this step.

STEPLIB DD
Defines the program library containing the DB2 load library.

SYSPRINT DD
Defines the data set for program messages.

UTPRINT DD
Defines the data set for DFSORT messages.

SYSUDUMP DD
Defines the dump data set for problem determination.

SORTOUT DD
Defines the data set for the SORT output.

SORTWKnn DD
Defines work data sets for data sorting; nn is a numeric.

ACCTDD DD
Defines the input data set for the ACCT LOAD utility.

STATDD DD
Defines the input data set for the STAT LOAD utility.

SUMMACCT DD
Defines the input data set for the ACCT SUMMARIZE LOAD utility.

DDFSACCT DD
Defines the input data set for the DDFACCT SUMMARIZE LOAD utility.

PKGSACCT DD
Defines the input data set for the PKGACCT SUMMARIZE LOAD utility.

BUFSACCT DD
Defines the input data set containing Buffer Summary data, for the BUFACCT SUMMARIZE LOAD utility.

AUDITDD DD
Defines the input data set for the AUDIT LOAD utility.

SYSIN DD

Defines the input data set containing the DB2 LOAD control statement.

## Print Statistics Report

STEP 5

Specifies the name of the Performance Reporter SMF report program as

PGM=DPREPORT

and the region required to run the program.

**Note:** The procedure symbolic RPTSTAT can be changed to IEFBR14 to disable this step.

STEPLIB DD

Defines the program library containing the DPREPORT load module.

DPDSTAT DD

Contains the statistics records created from the DB2 SMF 100 statistics records. This file is passed from Step 2 DPDSTAT.

The procedure symbolic DSDISP can be modified to catalog and keep this file for additional reporting.

SYSPRINT DD

Defines the data set for program messages.

DPDPRINT DD

Defines the data set for accounting or statistics reports.

SYSOUT DD

Defines the output class for sort messages.

SYSUDUMP DD

Defines the dump data set for problem determination.

SORTWKnn DD

Defines work data sets for data sorting; nn is a numeric.

SYSIN DD

Control statements are defined in Chapter 4, “SMF Reporting Facilities (DPREPORT)” on page 57.

As a default, Step 5 produces a statistics long report for the total interval included in the input data. The control statements can be modified to select data by date, time, relative day, SMFID, or DB2 ID. A different summarization interval can also be specified.

## Print Accounting Report

STEP 6

Specifies the name of the Performance Reporter SMF report program as

PGM=DPREPORT

and the region required to run the program.

**Note:** The procedure symbolic RPTACCT can be changed to IEFBR14 to disable this step.

STEPLIB DD

Defines the program library containing the DPREPORT load module.

DPDACCT DD

Contains the accounting records created from the DB2 SMF 101 accounting records. This file is passed from Step 1 DPDACCT.

The procedure symbolic DSDISP can be modified to catalog and keep this file for additional reporting.

SYSPRINT DD

Defines the data set for program messages.

DPDPRINT DD

Defines the data set for accounting or statistics reports.

SYSOUT DD

Defines the output class for sort messages.

SYSUDUMP DD

Defines the dump data set for problem determination.

SORTWKnn DD

Defines work data sets for data sorting; nn is a numeric.

SYSIN DD

Control statements are defined in Chapter 4, “SMF Reporting Facilities (DPREPORT)” on page 57.

As a default, Step 6 produces an accounting long report by plan for the total interval included in the input data. The control statements can be modified to select data by date, time, relative day, SMFID, DB2 ID, or various thread identifiers. The summarization keys and interval can also be modified.

Sample job control statements for the six-step SMF data extraction process are shown in Figure 3 on page 20.

```
//DPRSMF   JOB
//*------------------------------------------------------------*
//*       JOB TO EXTRACT DB2-RELATED RECORDS FROM              *
//*       SMF, AND LOAD THEM INTO DMR PERFORMANCE              *
//*       REPORTER TABLES FOR REPORTING                        *
//*------------------------------------------------------------*
//DPPSMF PROC HIDP='HILVL.RUN.LIB',              <-- UPDATE
//             HIDB2=DSN410,                      <-- VERIFY
//             SMF='SYS3.SMF.DAILY(0)',           <-- SMF DATASET
//             UNIT=SYSDA,                        <-- VERIFY
//*                     THE FOLLOWING IS FOR DB2 LOAD UTILITY
//             SYSTEM=DDDD,                       <-- UPDATE
//             UID=ANYUTID,                       <-- UPDATE
//*                     THE FOLLOWING IS FOR REPORTS FROM SMF
//             RPTHI=HILVL,                       <-- UPDATE
//             RPTUNIT=SYSDA,                     <-- VERIFY
//             RPTSTAT=DPREPORT,                  <-- OR IEFBR14
//             RPTACCT=DPREPORT,                  <-- OR IEFBR14
//             DSSTAT=NEW,                        <-- OR OLD STAT
//             DSDISP=DELETE,                     <-- OR CATLG DISP
//*                     THE FOLLOWING CAN DISABLE SUM / LOAD
//             SUMPGM=DPSUMLD,                    <-- OR IEFBR14
//             LOADPGM=DSNUTILB                   <-- OR IEFBR14
//*------------------------------------------------------------*
//*            EXTRACT 100 AND 101 RECORDS FROM SMF,           *
//*            SORT, AND REFORMAT TO DPDACCT AND DPDSTAT       *
//*            DATASETS. OPTIONALLY EXTRACT 102 AUDIT          *
//*            RECORDS, SORT AND REFORMAT TO DPDAUDIT          *
//*------------------------------------------------------------*
//*
//STEP1   EXEC PGM=DPRDSMF,REGION=4096K
//STEPLIB   DD DISP=SHR,DSN=&HIDP..BBLINK
//DPRLOG    DD SYSOUT=*              SMF EXTRACT LOG
//SYSOUT    DD SYSOUT=*              SORT MESSAGES
//SORTMSG   DD SYSOUT=*              SORT MESSAGES
//SYSUDUMP DD SYSOUT=*
//DPDACCT   DD DSN=&RPTHI..DPDACCT, INTERIM ACCTG FILE FOR LOAD/RPTS
//             DISP=(&DSSTAT,PASS),
//             UNIT=&RPTUNIT,
//             SPACE=(CYL,(800,200),RLSE),
//             DCB=BLKSIZE=23760
//DPDSTAT   DD DISP=(,PASS),UNIT=&UNIT,SPACE=(CYL,(20,10),RLSE),
//             DCB=BLKSIZE=23760
//DPDAUDIT DD DISP=(,PASS),UNIT=&UNIT,SPACE=(CYL,(200,100),RLSE),
//             DCB=BLKSIZE=23760
//SORTIN    DD DISP=SHR,DSN=&SMF
//SORTWK01 DD UNIT=&UNIT,SPACE=(CYL,(100,50))
//SORTWK02 DD UNIT=&UNIT,SPACE=(CYL,(100,50))
//SORTWK03 DD UNIT=&UNIT,SPACE=(CYL,(100,50))
//SORTWK04 DD UNIT=&UNIT,SPACE=(CYL,(100,50))
//SORTWK05 DD UNIT=&UNIT,SPACE=(CYL,(100,50))
//SORTWK06 DD UNIT=&UNIT,SPACE=(CYL,(100,50))
//DPRCNTL   DD DUMMY           CONTROL STATEMENTS
```

Figure 3.  Sample DPRSMF Job Control Statements (Page 1 of 7)

```
//*------------------------------------------------------------*
//*            CHANGE STATS RECORDS FROM DB2 ACCUMULATED VALUES*
//*            TO INTERVAL (DELTA) VALUES.                     *
//**************************************************************
//*****WARNING - THIS STEP MUST BE RUN FOR STATISTICS **********
//*****WARNING - PROCESSING OR ELSE VERY LARGE NUMBERS**********
//*****WARNING - IN STATISTICS REPORTS WILL OCCUR     **********
//**************************************************************
//*------------------------------------------------------------*
//*
//STEP2   EXEC PGM=DPRSTAT,REGION=4096K
//STEPLIB   DD DISP=SHR,DSN=&HIDP..BBLINK
//DPRLOG    DD SYSOUT=*
//SYSOUT    DD SYSOUT=*              SORT MESSAGES
//SORTMSG   DD SYSOUT=*              SORT MESSAGES
//SYSUDUMP DD SYSOUT=*
//SORTIN    DD DISP=(OLD,DELETE),DSN=*.STEP1.DPDSTAT
//          DD DISP=OLD,DSN=&HIDP..SPIN READ OLD SPIN DATA
//DPDSPIN   DD DISP=OLD,DSN=&HIDP..SPIN WRITE NEW SPIN DATA
//DPDSTAT   DD DSN=&RPTHI..DPDSTAT, INTERIM STATS FILE FOR LOAD/RPTS
//             DISP=(&DSSTAT,PASS),
//             UNIT=&RPTUNIT,
//             SPACE=(CYL,(5,1),RLSE),
//             DCB=BLKSIZE=23760
//SORTWK01 DD UNIT=&UNIT,SPACE=(CYL,(2,2))
//SORTWK02 DD UNIT=&UNIT,SPACE=(CYL,(2,2))
//SORTWK03 DD UNIT=&UNIT,SPACE=(CYL,(2,2))
//*
//*------------------------------------------------------------*
//*       BUILD DB2 LOAD CONTROL STATEMENTS,                   *
//*       SUMMARIZE ACCOUNTING AND DDF ACCOUNTING RECORDS,     *
//*       PURGE OLD RECORDS                                    *
//*------------------------------------------------------------*
//*
//STEP3   EXEC PGM=&SUMPGM,REGION=4096K          DPSUMLD OR IEFBR14
//STEPLIB   DD DISP=SHR,DSN=&HIDB2..SDSNLOAD
//          DD DISP=SHR,DSN=&HIDP..BBLINK
//DPSYSOUT DD SYSOUT=*            PROGRAM MESSAGES
//SYSOUT    DD SYSOUT=*            SORT MESSAGE
//SORTMSG   DD SYSOUT=*              SORT MESSAGES
//SYSUDUMP DD SYSOUT=*
//SORTWK01 DD UNIT=&UNIT,SPACE=(CYL,(100,50))
//SORTWK02 DD UNIT=&UNIT,SPACE=(CYL,(100,50))
//SORTWK03 DD UNIT=&UNIT,SPACE=(CYL,(100,50))
//SORTWK04 DD UNIT=&UNIT,SPACE=(CYL,(100,50))
//SORTWK05 DD UNIT=&UNIT,SPACE=(CYL,(100,50))
//SORTWK06 DD UNIT=&UNIT,SPACE=(CYL,(100,50))
//DPSYSIN   DD DUMMY            CONTROL STATEMENTS
//DPLOAD    DD UNIT=&UNIT,DISP=(,PASS),SPACE=(TRK,(1,1),RLSE),
//             DCB=(RECFM=FB,LRECL=80,BLKSIZE=3200)
//*                  USER-DEFINED DD STATEMENTS
//ACCTDD    DD DISP=(OLD,PASS),DSN=&RPTHI..DPDACCT
//SUMMACCT DD UNIT=&UNIT,DISP=(,PASS),SPACE=(CYL,(600,300),RLSE),
//             DCB=(LRECL=2048,BLKSIZE=23760,RECFM=VB)
//DDFSACCT DD UNIT=&UNIT,DISP=(,PASS),SPACE=(CYL,(10,2),RLSE),
//             DCB=(LRECL=2048,BLKSIZE=23760,RECFM=VB)
//PKGSACCT DD UNIT=&UNIT,DISP=(,PASS),SPACE=(CYL,(10,2),RLSE),
//             DCB=(LRECL=2048,BLKSIZE=23760,RECFM=VB)
//BUFSACCT DD UNIT=&UNIT,DISP=(,PASS),SPACE=(CYL,(10,2),RLSE),
//             DCB=(LRECL=2048,BLKSIZE=23760,RECFM=VB)
```

Figure 3. Sample DPRSMF Job Control Statements (Page 2 of 7)

```
//*------------------------------------------------------------*
//*            LOAD INTO DB2 TABLES                            *
//*------------------------------------------------------------*
//*
//STEP4   EXEC PGM=&LOADPGM,REGION=4096K,      DSNUTILB OR IEFBR14
//             PARM='&SYSTEM,&UID'
//STEPLIB   DD DSN=&HIDB2..SDSNLOAD,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SORTMSG   DD SYSOUT=*              SORT MESSAGES
//UTPRINT   DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SORTOUT   DD UNIT=&UNIT,SPACE=(CYL,(10,2))
//SORTWK01 DD UNIT=&UNIT,SPACE=(CYL,(100,50))
//SORTWK02 DD UNIT=&UNIT,SPACE=(CYL,(100,50))
//SORTWK03 DD UNIT=&UNIT,SPACE=(CYL,(100,50))
//SORTWK04 DD UNIT=&UNIT,SPACE=(CYL,(100,50))
//SORTWK05 DD UNIT=&UNIT,SPACE=(CYL,(100,50))
//SORTWK06 DD UNIT=&UNIT,SPACE=(CYL,(100,50))
//SYSUT1    DD UNIT=&UNIT,SPACE=(CYL,(5,2))
//SYSIN     DD DISP=(OLD,DELETE),DSN=*.STEP3.DPLOAD
//*
//ACCTDD    DD DISP=(OLD,PASS),DSN=&RPTHI..DPDACCT
//STATDD    DD DISP=(OLD,PASS),DSN=&RPTHI..DPDSTAT
//SUMMACCT  DD DISP=(OLD,DELETE),DSN=*.STEP3.SUMMACCT
//DDFSACCT  DD DISP=(OLD,DELETE),DSN=*.STEP3.DDFSACCT
//PKGSACCT  DD DISP=(OLD,DELETE),DSN=*.STEP3.PKGSACCT
//BUFSACCT  DD DISP=(OLD,DELETE),DSN=*.STEP3.BUFSACCT
//AUDITDD   DD DISP=(OLD,DELETE),DSN=*.STEP1.DPDAUDIT
//*------------------------------------------------------------*
//*            PRINT STATISTICS REPORT (LONG OR SHORT)         *
//*------------------------------------------------------------*
//*
//STEP5    EXEC PGM=&RPTSTAT,REGION=4096K      DPREPORT OR IEFBR14
//STEPLIB  DD   DISP=SHR,DSN=&HIDP..BBLINK
//DPDSTAT  DD  DSN=&RPTHI..DPDSTAT,DISP=(OLD,&DSDISP) DELETE OR CATLG
//SYSPRINT DD   SYSOUT=*
//DPDPRINT DD   SYSOUT=*
//SYSOUT   DD   SYSOUT=*
//SYSUDUMP DD   SYSOUT=*
//SORTWK01 DD   UNIT=&UNIT,SPACE=(TRK,(45,15)),DISP=(,DELETE)
//SORTWK02 DD   UNIT=&UNIT,SPACE=(TRK,(45,15)),DISP=(,DELETE)
//SORTWK03 DD   UNIT=&UNIT,SPACE=(TRK,(45,15)),DISP=(,DELETE)
//SORTWK04 DD   UNIT=&UNIT,SPACE=(TRK,(45,15)),DISP=(,DELETE)
//SORTWK05 DD   UNIT=&UNIT,SPACE=(TRK,(45,15)),DISP=(,DELETE)
//SORTWK06 DD   UNIT=&UNIT,SPACE=(TRK,(45,15)),DISP=(,DELETE)
//SORTWK07 DD   UNIT=&UNIT,SPACE=(TRK,(45,15)),DISP=(,DELETE)
//SORTWK08 DD   UNIT=&UNIT,SPACE=(TRK,(45,15)),DISP=(,DELETE)
//*
//*  CONTROL STATEMENTS FOR STATISTICS REPORT
//*    -- REPTYPE=LONG OR SHORT
//*    -- INTVL=NN (MINUTES) OR DEFAULT TO COMPLETE INPUT INTERVAL
//*   -- SEE MANUAL FOR OTHER KEYWORDS FOR DAY / TIME / DB2 SELECTION
//*
//SYSIN    DD DUMMY
```

Figure 3.  Sample DPRSMF Job Control Statements (Page 3 of 7)

```
//*-----------------------------------------------------------*
//*            PRINT ACCOUNTING REPORT (LONG OR SHORT)         *
//*-----------------------------------------------------------*
//*
//STEP6    EXEC PGM=&RPTACCT,REGION=4096K      DPREPORT OR IEFBR14
//STEPLIB DD   DISP=SHR,DSN=&HIDP..BBLINK
//DPDACCT  DD  DSN=&RPTHI..DPDACCT,DISP=(OLD,&DSDISP) DELETE OR CATLG
//SYSPRINT DD   SYSOUT=*
//DPDPRINT DD   SYSOUT=*
//SYSOUT   DD   SYSOUT=*
//SYSUDUMP DD   SYSOUT=*
//SORTWK01 DD   UNIT=&UNIT,SPACE=(TRK,(45,15)),DISP=(,DELETE)
//SORTWK02 DD   UNIT=&UNIT,SPACE=(TRK,(45,15)),DISP=(,DELETE)
//SORTWK03 DD   UNIT=&UNIT,SPACE=(TRK,(45,15)),DISP=(,DELETE)
//SORTWK04 DD   UNIT=&UNIT,SPACE=(TRK,(45,15)),DISP=(,DELETE)
//SORTWK05 DD   UNIT=&UNIT,SPACE=(TRK,(45,15)),DISP=(,DELETE)
//SORTWK06 DD   UNIT=&UNIT,SPACE=(TRK,(45,15)),DISP=(,DELETE)
//SORTWK07 DD   UNIT=&UNIT,SPACE=(TRK,(45,15)),DISP=(,DELETE)
//SORTWK08 DD   UNIT=&UNIT,SPACE=(TRK,(45,15)),DISP=(,DELETE)
//*
//*  CONTROL STATEMENTS FOR ACCOUNTING REPORT
//*    -- REPTYPE=LONG OR SHORT
//*    -- INTVL=NN (MINUTES) OR DEFAULT TO COMPLETE INPUT INTERVAL
//*    -- RORDER DEFINES THE SUMMARY KEYS
//*       (PRAUTH,PLANAM,CONNID,CORRID,ORAUTH,REQLOC,CONTYP,PAKAGE)
//*    -- FILTYP/FILTR1/FILTR2 DEFINE WORKLOAD SELECTION
//*    -- SEE MANUAL FOR OTHER KEYWORDS AND DAY / TIME / DB2 SELECTION
//*
//SYSIN    DD DUMMY
//        PEND
//*
//JS010   EXEC DPPSMF
//STEP1.DPRCNTL DD *
   TYPE=(ACCT,DDFACCT,PKGACCT,BUFACCT)
   TYPE=(STAT,DDFSTAT,BUFSTAT)
   TYPE2=(AUDSUM,AUDFAIL,AUDGRV,AUDDDL,AUDDML,AUDDMB,AUDCHG,AUDUTL)
//*
```

Figure 3. Sample DPRSMF Job Control Statements (Page 4 of 7)

```
//STEP3.DPSYSIN DD *
   GLOBAL LOG=NO,SSID=DDDD,INVFIELD=IGNORE,PLAN=DPSUMLD <-- UPDATE
   LOAD   TYPE=STAT,TABLE=DMRPR.DMRSTAT,FROMDD=STATDD
   LOAD   TYPE=DDFSTAT,TABLE=DMRPR.DMRSTDF,FROMDD=STATDD
   LOAD   TYPE=BUFSTAT,TABLE=DMRPR.DMRSBFDT,FROMDD=STATDD
*------------------------------------------------------------------*
*             THE FOLLOWING STATEMENTS ARE COMMENTED TO AVOID      *
*             THE OVERHEAD OF LOADING THE DETAIL ACCOUNTING TABLES.*
*             REMOVE THE COMMENTS IF THE DETAIL ACCOUNTING TABLES *
*             ARE DESIRED.                                         *
* LOAD   TYPE=ACCT,TABLE=DMRPR.DMRACDTL,FROMDD=ACCTDD             *
* LOAD   TYPE=DDFACCT,TABLE=DMRPR.DMRADDTL,FROMDD=ACCTDD          *
* LOAD   TYPE=PKGACCT,TABLE=DMRPR.DMRAPDTL,FROMDD=ACCTDD          *
* LOAD   TYPE=BUFACCT,TABLE=DMRPR.DMRABDTL,FROMDD=ACCTDD          *
* PURGE TYPE=ACCT,TABLE=DMRPR.DMRACDTL,RETPD=10D                 *
* PURGE TYPE=DDFACCT,TABLE=DMRPR.DMRADDTL,RETPD=10D              *
* PURGE TYPE=PKGACCT,TABLE=DMRPR.DMRAPDTL,RETPD=10D              *
* PURGE TYPE=BUFACCT,TABLE=DMRPR.DMRABDTL,RETPD=10D              *
*------------------------------------------------------------------*
*
*         --- LOAD SUMMARY ACCOUNTING TABLE (30-MINUTE INTERVAL) ---
*         --- PURGE DATA THAT IS OLDER THAN 10 DAYS ---
*
   SUMMARIZE TYPE=SUMACCT,FROMDD=ACCTDD,TABLE=DMRPR.DMRACSUM,
       TODD=SUMMACCT,
       SUMMSTART=*_1,
       SUMMEND=*,
       SUMMINT=30M,
       COMBINE=(CREATETBL,CREATEINDX,CREATETSP,CREATESYN,CREATEDB,
                CREATESG,CREATEVU,CREATEALIAS),
       COMBINE=(ALTERSG,ALTERTSP,ALTERTBL,ALTERINDX,ALTERDB),
       COMBINE=(DROPINDX,DROPTBL,DROPTSP,DROPDB,DROPSYN,DROPSG,
                DROPVU,DROPALIAS,DROPPKG),
       SUMMKEY=(LOCATION,SUBSYSTEM,PLANNAME,AUTHID)
*------------------------------------------------------------------*
*      OTHER POSSIBLE SUMMARY KEY VALUES ARE CONNECTION, CORRID    *
*      ORIGPRIMID, LUWIDNID, LUWIDLUNM. *
*------------------------------------------------------------------*
   LOAD   TYPE=SUMACCT,TABLE=DMRPR.DMRACSUM,FROMDD=SUMMACCT
   PURGE TYPE=SUMACCT,TABLE=DMRPR.DMRACSUM,RETPD=10D
*
*        --- LOAD SUMMARY DDF ACCOUNTING TABLE (30-MINUTE INTERVAL)-
*         --- PURGE DATA THAT IS OLDER THAN 10 DAYS ---
*
   SUMMARIZE TYPE=SUMDDFACCT,FROMDD=ACCTDD,TABLE=DMRPR.DMRADSUM,
       SUMMSTART=*_1,
       SUMMEND=*,
       SUMMINT=30M,
       TODD=DDFSACCT,
       SUMMKEY=(LOCATION,SUBSYSTEM,PLANNAME,AUTHID,DDFLOCATION)
```

Figure 3. Sample DPRSMF Job Control Statements (Page 5 of 7)

```
*------------------------------------------------------------------*
*      OTHER POSSIBLE SUMMARY KEY VALUES ARE CONNECTION, CORRID    *
*      ORIGPRIMID, LUWIDNID, LUWIDLUNM, DDFLOCATION.               *
*------------------------------------------------------------------*
   LOAD   TYPE=SUMDDFACCT,TABLE=DMRPR.DMRADSUM,FROMDD=DDFSACCT
   PURGE TYPE=SUMDDFACCT,TABLE=DMRPR.DMRADSUM,RETPD=10D
*
*         --- LOAD SUMMARY PKG ACCOUNTING TABLE (30-MINUTE INTERVAL)-
*         --- PURGE DATA THAT IS OLDER THAN 10 DAYS ---
*
   SUMMARIZE TYPE=SUMPKGACCT,FROMDD=ACCTDD,TABLE=DMRPR.DMRAPSUM,
       SUMMSTART=*_1,
       SUMMEND=*,
       SUMMINT=30M,
       TODD=PKGSACCT,
       SUMMKEY=(LOCATION,SUBSYSTEM,PLANNAME,AUTHID,PROGRAMNAME)
*------------------------------------------------------------------*
*      OTHER POSSIBLE SUMMARY KEY VALUES ARE CONNECTION, CORRID    *
*      ORIGPRIMID, COLLECTIONID.                                   *
*------------------------------------------------------------------*
   LOAD   TYPE=SUMPKGACCT,TABLE=DMRPR.DMRAPSUM,FROMDD=PKGSACCT
   PURGE TYPE=SUMPKGACCT,TABLE=DMRPR.DMRAPSUM,RETPD=10D
*
   SUMMARIZE TYPE=SUMBUFACCT,FROMDD=ACCTDD,TABLE=DMRPR.DMRABSUM,
       SUMMSTART=*_1,
       SUMMEND=*,
       SUMMINT=30M,
       TODD=BUFSACCT,
       SUMMKEY=(LOCATION,SUBSYSTEM,PLANNAME,AUTHID,BPNAME)
*------------------------------------------------------------------*
*      OTHER POSSIBLE SUMMARY KEY VALUES ARE CONNECTION, CORRID    *
*      ORIGPRIMID, COLLECTIONID.                                   *
*------------------------------------------------------------------*
   LOAD   TYPE=SUMBUFACCT,TABLE=DMRPR.DMRABSUM,FROMDD=BUFSACCT
   PURGE TYPE=SUMBUFACCT,TABLE=DMRPR.DMRABSUM,RETPD=10D
```

Figure 3. Sample DPRSMF Job Control Statements (Page 6 of 7)

```
*------------------------------------------------------------------*
*             THE FOLLOWING STATEMENTS WILL LOAD AUDIT TABLES.    *
*------------------------------------------------------------------*
   LOAD TYPE2=AUDSUM,TABLE=DMRPR.DMRAUSUM,FROMDD=AUDITDD
   LOAD TYPE2=AUDFAIL,TABLE=DMRPR.DMRAUFAL,FROMDD=AUDITDD
   LOAD TYPE2=AUDGRV,TABLE=DMRPR.DMRAUGRV,FROMDD=AUDITDD
   LOAD TYPE2=AUDDDL,TABLE=DMRPR.DMRAUDDL,FROMDD=AUDITDD
   LOAD TYPE2=AUDDML,TABLE=DMRPR.DMRAUDML,FROMDD=AUDITDD
   LOAD TYPE2=AUDDMB,TABLE=DMRPR.DMRAUDMB,FROMDD=AUDITDD
   LOAD TYPE2=AUDCHG,TABLE=DMRPR.DMRAUCHG,FROMDD=AUDITDD
   LOAD TYPE2=AUDUTL,TABLE=DMRPR.DMRAUUTL,FROMDD=AUDITDD
   PURGE TYPE2=AUDSUM,TABLE=DMRPR.DMRAUSUM,RETPD=10D
   PURGE TYPE2=AUDFAIL,TABLE=DMRPR.DMRAUFAL,RETPD=10D
   PURGE TYPE2=AUDGRV,TABLE=DMRPR.DMRAUGRV,RETPD=10D
   PURGE TYPE2=AUDDDL,TABLE=DMRPR.DMRAUDDL,RETPD=10D
   PURGE TYPE2=AUDDML,TABLE=DMRPR.DMRAUDML,RETPD=10D
   PURGE TYPE2=AUDDMB,TABLE=DMRPR.DMRAUDMB,RETPD=10D
   PURGE TYPE2=AUDCHG,TABLE=DMRPR.DMRAUCHG,RETPD=10D
   PURGE TYPE2=AUDUTL,TABLE=DMRPR.DMRAUUTL,RETPD=10D
*------------------------------------------------------------------*
//STEP5.SYSIN DD *
REPORT=STAT
REPTYP=LONG
/*
//STEP6.SYSIN DD *
REPORT=ACCT
REPTYP=LONG
RORDER=(PLANAM)
/*
//
```

Figure 3.  Sample DPRSMF Job Control Statements (Page 7 of 7)

## DPSUMLD Control Statements

The DPSUMLD program is used in the DPRSMF job when loading data into tables from SMF and in the DPRSUM job when processing data already in the tables.

The usage and meaning of some of the keywords and operands are slightly different in the two environments. For example, of the table types defined in the TYPE keyword, the detail tables (ACCT, STAT, BUFFER, DDFACCT, DDFSTAT) can only be loaded from SMF data in the DPRSMF job, while the summary tables (SUMACCT and SUMDDFACCT) can be summarized and loaded in both—from SMF data in DPRSMF, or from another detail or summary table in DPRSUM.

## Control Statement Overview

The following control statements are valid:

GLOBAL
Global defaults. One is required as the first statement.

LOAD
Load input data in a temporary file to a table. Specify one per output table.

In the DPRSMF job, this statement is used to load processed SMF input data from a temporary file to a detail output table, or after a SUMMARIZE statement to load summarized SMF input data from a work file to an output summary table.

In the DPRSUM job, this statement is used after a SUMMARIZE statement to load summarized data from a work file to an output summary table.

SUMMARIZE
Summarize input data to a work file. This statement must be followed by a LOAD statement. Specify one per summary output table (valid only for accounting tables).

In the DPRSMF job, this statement is used to summarize SMF input data.

In the DPRSUM job, this statement is used to summarize data in a table. The input table may be either a detail table or a summary table that is to be even further summarized.

PURGE
Purge data that is older than the specified datetime from a table. Specify one statement per table where purging is required.

In the DPRSMF job, this statement is used to purge table data daily; for example, deleting the data of the oldest day in the table while loading the data from the previous work day.

In the DPRSUM job, this statement is used to purge table data less frequently, during a summarization job cycle that may be run only weekly or monthly.

UNLOAD
Unload data from a table to a data set. This statement is used only for table migration; for example, to unload data from a table where all possible columns are defined to a smaller table with only a subset of columns.

## Control Statement Detail Definitions

The control statements have the following operands:

- **GLOBAL**

  INVFIELD=IGNORE|ABORT
  Defines if processing should continue if an invalid field in the DB2 table is encountered. The default is ABORT, which indicates that the processing for the request should abort.

  LOG=YES|NO
  Defines the default logging control for all load requests that follow. The default is NO. This statement can occur multiple times. It resets the default value for all statements that follow.

  SSID=ssid of target DB2 system
  Subsystem name of the DB2 system which owns the tables to be loaded, summarized or purged. This statement can occur multiple times. It resets the default value for all statements that follow. This is a required field with no default.

  PLAN=name of the plan
  Name of the plan for the load interface routine. This is provided in the case where you wish to change the default name of the plan. If not provided the default plan as defined in the installation instructions is used.

- **LOAD**

  Requests that the defined input be loaded to the defined output table by the DB2 LOAD utility.

  TYPE=type of accounting or statistics table
  Specifies the type of accounting or statistics table. The possible values are

  – ACCT
  – SUMACCT
  – DDFACCT
  – SUMDDFACCT
  – PKGACCT
  – SUMPKGACCT
  – BUFACCT
  – SUMBUFACCT
  – STAT
  – DDFSTAT
  – BUFSTAT

  This is a required field with no default.

TYPE2=type of audit table

Specifies the type of audit table. The possible values are

- AUDFAIL
- AUDGRV
- AUDDDL
- AUDDML
- AUDDMB
- AUDCHG
- AUDUTL
- AUDSUM

This is a required field with no default.

TABLE=tablename

Name of the receiving table. This table is read to define the columns defined within the table. This is a required field with no default.

LOG=YES|NO

Defines the LOG parameter for the load of this table. If not defined, the default is defined by the GLOBAL LOG value.

FROMDD=ddname

Ddname assigned to the file that is to be loaded. This is a required field with no default.

REPLACE=YES|NO

Specifies whether the table space and all its indexes need to be reset to empty before records are loaded.

The default is NO; however, if KEEPDICTIONARY=YES is specified, REPLACE= YES becomes the default.

KEEPDICTIONARY=YES|NO

YES prevents the LOAD utility from building a new compression dictionary. LOAD retains the current compression dictionary and uses it for compressing the input data. This option eliminates the cost associated with building a new dictionary. The default is NO.

**Note:** This keyword is valid only if a compression dictionary exists and the table space being loaded has the COMPRESS YES attribute.

SORTKEYS=nnnnn

Is an estimate of the number of index keys to be sorted in parallel with the reload and build phases to improve performance. The default is 0.

- **SUMMARIZE**

Requests that the input values be summarized according to the included specifications.

TYPE=type of table

Specifies the type of the receiving table. The possible values are

- SUMACCT
- SUMDDFACCT
- SUMPKGACCT
- SUMBUFACCT

This is a required field with no default.

TABLE=tablename
Name of the receiving table. This table is read to define the columns defined within the table. It can be loaded by a specifically requested LOAD request. This is an optional field. If it is not specified, there is no source of default summarization columns, and SUMMKEY is the only other possible source.

FROMTBL=tablename
Name of the table from which records are to be read and summarized. This is mutually exclusive with FROMDD.

FROMDD=ddname
Ddname assigned to the file that is to be loaded. This is mutually exclusive with FROMTBL.

TODD=ddname
Ddname receiving the file of summarized records. This is used as a work file for the specified LOAD request. This is a required field with no default.

COMBINE=(fld1,fld2,fld3,...)
Defines a list of column names from the input table which are to be combined in the output table. The values from all fields are added together and passed on in the first field.

It is possible to type multiple COMBINE statements in a single summarization process.

There is no default. The default is to ignore combine processing.

Only columns with like attributes can be combined. It is possible to combine integer and short integer columns. Character columns cannot be combined, nor can datetime columns.

If overflow occurs while combining fields, the largest possible positive value is placed in the field.

**Note:** There is a limit of 200 field names in all COMBINE statements for a summarization process.

SUMMSTART=date-time
Defines the first record which participates in summarization.

Where date is

| | |
|---|---|
| yyyy-mm-dd | For a specific date |
| * | For the current date |
| *_nn | For a previous date nn days before the current date |
| *_l | For the beginning of the previous day |

and time is

| | |
|---|---|
| hh:mm:ss | For a specific time |

If omitted, the default is to start with the oldest record from the specified input source.

**Note:** For ease in reading the resulting reports, it is preferable to specify a start time on an even time period, such as an hour.

SUMMEND=date-time

Defines the last record which participates in summarization.

Where date is

| | |
|---|---|
| yyyy-mm-dd | Up to (but not including) the specified date |
| * | Up to (but not including) the current date |
| *_nn | Up to (but not including) a previous date nn days before the current date |

and time is

| | |
|---|---|
| hh:mm:ss | Up to (but not including) the specified time |

If omitted, the default is to process all records after the first one selected.

SUMMINT=nnnX

Indicates the number of minutes (M), hours (H), days (D), weeks (W), or months (N or MON) to be included in each summary interval (row).

The X can be entered as M, H, D, W, N, or MON.

The default value is 30M.

SUMMKEY=(key,key,...)

Defines which key values are to remain unique in the summarized values. The possible values are

- LOCATION
- SUBSYSTEM
- PLANNAME
- AUTHID
- CONNECTION
- CORRID
- ORIGPRIMID
- LUWIDNID
- LUWIDLUNM
- DDFLOCATION
- EXECLOCATION
- COLLECTIONID
- PROGRAMNAME
- CONSISTOKEN
- BPNAME

Any combination of these values can be entered separated by commas. All values entered must be defined in the receiving table, if the receiving table is named. The SUMMKEY or TABLE keyword (or both) must be defined in a SUMMARIZE statement. Some limitations are:

| **If the TYPE of table is...** | **SUMACCT** | **SUMDDFACCT** | **SUMPKGACCT** | **SUMBUFACCT** |
|---|---|---|---|---|
| **Key Values** | **Then the possible values are...** | | | |
| LOCATION | X | X | X | X |
| SUBSYSTEM | X | X | X | X |
| PLANNANME | X | X | X | X |
| AUTHID | X | X | X | X |
| CONNECTION | X | X | X | X |
| CORRID | X | X | X | X |
| ORIGPRIMID | X | X | X | X |
| LUWIDNID | X | X | | |
| LUWIDLUNM | X | X | | |
| DDFLOCATION | | X | | |
| EXECLOCATION | | | X | |
| COLLECTIONID | | | X | |
| PROGRAMNAME | | | X | |
| CONSISTOKEN | | | X | |
| BPNAME | | | | X |

The default is to use all key columns that are defined in the receiving table.

- **PURGE**

Requests that the specified table be purged of all data with a datetime older that the specified value.

TYPE=type of table
Specifies the type of table. The possible values are

- ACCT
- SUMACCT
- DDFACCT
- SUMDDFACCT
- PKGACCT
- SUMPKGACCT
- BUFACCT
- SUMBUFACCT
- STAT
- DDFSTAT
- BUFSTAT

This field is included only for compatibility with other control statements. In the present release it is for documentation only.

TABLE=tablename
Name of the table to be processed. This is a required field with no default.

EXPDT=yyyy-mm-dd-hh:mm:ss
Timedate of the oldest entry that is to be kept. All older entries are deleted. This is mutually exclusive with RETPD.

RETPD=nnnD
Interval describing the oldest entry that is to be kept. Any entry older than today minus nnn days is deleted. This is mutually exclusive with EXPDT.

RETPD=0D deletes all rows in the selected table.

- **UNLOAD**

  Requests that all the rows from the DB2 table are to be written to a variable block sequential data set.

  TYPE=type of accounting or statistics table
  Specifies the type of accounting or statistics table. The possible values are

  - ACCT
  - SUMACCT
  - DDFACCT
  - SUMDDFACCT
  - PKGACCT
  - SUMPKGACCT
  - BUFACCT
  - SUMBUFACCT
  - STAT
  - DDFSTAT
  - BUFSTAT

  This is a required field with no default.

  TYPE2=type of audit table
  Specifies the type of audit table. The possible values are

  - AUDFAIL
  - AUDGRV
  - AUDDDL
  - AUDDML
  - AUDDMB
  - AUDCHG
  - AUDUTL
  - AUDSUM

  This is a required field with no default.

  FROMTBL=tablename
  Name of the table from which the rows are fetched. This is a required field with no default.

  TODD=ddname
  Ddname assigned to the file to which the rows are written. This is a required field with no default.

UNLDSTART=yyyy-mm-dd,UNLDEND=yyyy-mm-dd

Limits the records that are unloaded; for example:

```
UNLDSTART=2001-01-08,UNLDEND=2001-01-09
```

limits the records unloaded to those for the day of 2001-01-08.

## Example of Control Statements

Figure 4 is an example of the control statements showing proper syntax and structure for a single step.

```
    GLOBAL    LOG=NO,SSID=PROD,PLAN=DZNEWPL
*
*   --------- Detail Accounting ---------
    LOAD      TYPE=ACCT,TABLE=DMRPR.ACCT1,
              FROMDD=DDACCT
*
*   --------- Statistics ----------------
    LOAD      TYPE=STAT,TABLE=DMRPR.STAT1,LOG=YES,
              FROMDD=DDSTAT
*
*   --------- DDF Accounting Summary, 2 hour intervals ----------
    SUMMARIZE TYPE=SUMDDFACCT,TABLE=DMRPR.DDFSAC1,
              FROMDD=DDFACCT,TODD=WKDD1,
              SUMMSTART=1993-09-14-02:00:00,
              SUMMEND=1993-09-15-12:00:00,
              SUMMINT=2H,
            SUMMKEY=(LOCATION,SUBSYSTEM,PLANNAME,AUTHID,DDFLOCATION)
*
    LOAD      TYPE=SUMDDFACCT,TABLE=DMRPR.DDFSAC1,LOG=YES,
              FROMDD=WKDD1
*
*   --------- DDF Accounting Summary, 10 Minute Intervals -------
    SUMMARIZE TYPE=SUMDDFACCT,TABLE=DMRPR.DDFSAC2,
              FROMDD=DDFACCT,TODD=WKDD2,
              SUMMSTART=1993-09-14-02:00:00,
              SUMMEND=1993-09-15-12:00:00,
              SUMMINT=10M,
            SUMMKEY=(LOCATION,SUBSYSTEM,PLANNAME,AUTHID,DDFLOCATION)
*
    LOAD      TYPE=SUMDDFACCT,TABLE=DMRPR.DDFSAC2,LOG=YES,
              FROMDD=WKDD2
*
*   --------- Purge DDF Accounting Summary Data ------------
    PURGE     TYPE=SUMDDFACCT,TABLE=DMRPR.DDFSAC2,
              EXPDT=1993-09-01
```

Figure 4. Control Statements Examples

## DPRSMF Return Codes and Error Messages

The following return codes and messages are produced by the DPRSMF program:

### Return Codes

The return codes produced by the DPRDSMF program are

| | |
|---|---|
| **0** | Successful completion |
| **4** | Warning issued |
| **8** | At least one major function was not completed |
| **12** | Catastrophic, no functions completed |

### Messages

The following messages are produced by the table load DPRDSMF program. All messages are written to the DZPRPRT data set.

- DPR0020E - GETMAIN FAILED - INCREASE REGION

  REASON: Not enough storage was available.

  SYSTEM ACTION: The step is terminated.

  USER ACTION: Increase the region size for the failing step.

- DPR0021E - LOAD FAILED FOR JXLOAD - CHECK STEPLIBS

  REASON: LOAD of module JXLOAD failed.

  SYSTEM ACTION: The step is terminated.

  USER ACTION: Check STEPLIB DD libraries and add necessary BBLINK library.

- DPR0030E - PARALLEL ACE GETMAIN FAILED - INCREASE REGION

  REASON: Not enough storage was available.

  SYSTEM ACTION: The step is terminated.

  USER ACTION: Increase the region size for the failing step.

- DPR0101I - nnnn - counter

  where

  nnnn Is a number

  counter Is the text description of the counter

  REASON: A series of counts is written at end of processing to show actions taken during processing.

  SYSTEM ACTION: None.

  USER ACTION: None.

- DPR0102I - ERROR IN OPEN FOR ddn

  REASON: The ddn is the ddname of the file which failed to open.

  SYSTEM ACTION: The step is terminated.

  USER ACTION: Determine reason for failure. (For example, DD statement is missing.)

- DPR0103I - INVALID RETURN FROM CONTROL STMTS, RC=rc

  REASON: Errors were detected in processing the control statements. The processing is terminated.

  SYSTEM ACTION: The step is terminated.

  USER ACTION: Correct and resubmit.

- DPR0104I - NON-ZERO RETURN CODE FROM SORT: rc

  REASON: Error in sort processing. rc=16 is the non-zero return code passed by the sort.

  SYSTEM ACTION: The step is terminated.

  USER ACTION: Determine the reason for sort failure from the sort messages. Correct and resubmit.

- DPR0105I - **WARNING** NO OUTPUT FOR REQUESTED TYPE: nnnnnnnn

  where

  nnnnnnnn Is the type of accounting or statistics table requested on the TYPE= statement

  REASON: The TYPE= statement requested a TYPE, but no DB2 SMF data for that type was available from the SORTIN DD data set.

  SYSTEM ACTION: None. Processing continues.

  USER ACTION: Either remove the TYPE from the TYPE= statement, or recheck the SMF input data and/or DB2 TRACE parameters.

- DPR0121I - nnnn - counter

  where

  nnnn Is a number

  counter Is the text description of the counter

  REASON: A series of counts is written at end of processing to show actions taken during processing.

  SYSTEM ACTION: None.

  USER ACTION: None.

- DPR0122I - ERROR IN OPEN FOR ddn

  REASON: The ddn is the ddname of the file which failed to open.

  SYSTEM ACTION: Step is terminated.

  USER ACTION: Determine reason for failure. (For example, DD statement is missing.)

- DPR0124I - NON-ZERO RETURN CODE FROM SORT: rc

  REASON: Error in sort processing; rc=16 is the non-zero return code passed by the sort.

  SYSTEM ACTION: Step is terminated.

  USER ACTION: Determine the reason for sort failure from the sort messages. Correct and resubmit.

- DPR0211I - ERROR IN fld - desc

  REASON: Error found in timezone specification.

  The possible combinations of fld and desc are

  | **Field** | **Description** |
  |---|---|
  | TZSTART | date edit message |
  | NUMBER OF TZ ENTRIES | UNMATCHED ENTRY |
  | TIMEZONE | date edit message |
  | TIMEZONE | MUST START WITH W OR L |
  | NUMBER ENTRIES | LIMITED TO 40 TOTAL |
  | TIMEZONE | MISSING DOT BEFORE TIME |

  The possible date edit messages are

  – DATE IS NOT NUMERIC

  – DAY CANNOT BE ZERO

  – INVALID MONTH VALUE

  – DAY TOO LARGE FOR MONTH

  – INVALID FORMAT FOR DATE

  – INVALID TIME FIELD

  – DAY NOT NUMERIC

  – ZERO DAY IS INVALID

  – YEAR NOT NUMERIC

  SYSTEM ACTION: This timezone specification is not used.

  USER ACTION: Correct and resubmit.

- DPR0212I - DUPLICATE TIMEZONE ENTRIES FOR smfid

  REASON: Duplicate timezone entries found.

  SYSTEM ACTION: The second entry is discarded.

  USER ACTION: Correct and resubmit.

- DPR0213I - SPECIFIC ENTRY FOR smfid WITH UNQUALIFIED ENTRIES

  REASON: Some timezone entries were not qualified with an SMFID, others were entered with a qualifier.

  SYSTEM ACTION: No timezone qualifiers are used, the step is terminated.

  USER ACTION: Correct and resubmit.

- DPR0214I - START DATETIME MUST BE LESS THAN END

  REASON: UNLDEND time specified is prior to UNLDSTART time.

  SYSTEM ACTION: Processing ends.

  USER ACTION: Correct and resubmit.

- JXR3010C - INTERNAL ERROR INVALID ROUTINE CODE - code

  REASON: Internal processing error.

  SYSTEM ACTION: Processing ends for the input statement.

  USER ACTION: Contact BMC Software.

- JXR3011E - SYNTAX ERROR - error

  REASON: A syntax error was found in the input.

  where

  error = UNMATCHED PARENTHESIS
  NO CONTINUATION FOUND
  IMBEDDED LEFT PARENTHESIS
  EXTRA RIGHT PARENTHESIS
  NO ENDING QUOTE
  INVALID QUOTED STRING
  ODD NUMBER OF HEX DIGITS
  INVALID HEX DIGITS

  SYSTEM ACTION: Statement is discarded.

  USER ACTION: Correct and resubmit.

- JXR3012E - DUPLICATE KEY ENTERED - key

  REASON: The keyword *key* was entered twice within one control statement

  SYSTEM ACTION: Statement is discarded.

  USER ACTION: Correct and resubmit.

- JXR3013E - INVALID KEYWORD key

  REASON: The keyword *key* is not valid in this control statement.

  SYSTEM ACTION: Statement is discarded.

  USER ACTION: Correct and resubmit.

- JXR3014I - VALUE 'value' FOR INVALID KEY SKIPPED

  REASON: An invalid or duplicate keyword was detected which had an associated value. This value is skipped.

  SYSTEM ACTION: Statement is discarded.

  USER ACTION: Correct and resubmit.

- JXR3015E - DATA ERROR FOR key - error desc.

  REASON: The data entered for a keyword *key* failed an edit. The error description specifies which edit was failed. The possible values include

  - MISSING 2ND RANGE VALUE
  - MUST HAVE A VALUE
  - date edit messages
  - EXCEEDED NUMBER OF VALUES
  - INPUT DATA TOO LONG
  - INPUT DATA TOO SHORT
  - INVALID DATA FOR INPUT
  - INPUT IS NOT NUMERIC
  - VALUE NOT ALLOWED
  - VALUE REQUIRED
  - INVALID PARENTHESES

  SYSTEM ACTION: The value is ignored.

  USER ACTION: Correct and resubmit.

- JXR3016W - NULL KEY VALUE FOUND

  REASON: A null value was found in scanning for next keyword. This is usually from two adjacent commas.

  SYSTEM ACTION: Skip to next keyword. Processing continues.

  USER ACTION: None.

## DPSUMLD Return Codes and Error Messages

The following return codes and messages are produced by the DPSUMLD program:

## Return Codes

The return codes produced by the table load DPSUMLD program are

| | |
|---|---|
| **0** | Successful completion |
| **4** | Warning issued |
| **8** | At least one major function was not completed |
| **12** | Catastrophic, no functions completed |

All messages are written to the DPSYSOUT data set.

## Informational Messages

The following informational messages are produced by the table load DPSUMLD program. All Messages are written to the DPSYSOUT data set.

- DB2 CONNECTION ESTABLISHED

  REASON: The program has established DB2 connection.

  SYSTEM ACTION: Processing continues.

  USER ACTION: None.

- LOAD CONTROL STATEMENT BUILT IN DPLOAD DATA SET

  REASON: The program has written a DB2 load control statement to DPLOAD data set.

  SYSTEM ACTION: Processing continues.

  USER ACTION: None.

- PURGE SUCCESSFUL

  REASON: The program has completed the purge request successfully.

  SYSTEM ACTION: Processing continues.

  USER ACTION: None.

- SUMMARIZATION SUCCESSFUL

  REASON: The program has completed summarization process successfully.

  SYSTEM ACTION: Processing continues.

  USER ACTION: None.

## Warning Messages

The following warning messages are produced by the table load DPSUMLD program. All messages are written to the DPSYSOUT data set.

- DB2 WARNING MESSAGES RECEIVED

  REASON: DB2 has issued warning messages.

  SYSTEM ACTION: Processing continues.

  USER ACTION: Refer to the warning message issued by DB2.

- PROCESSING CONTINUES AS INVFIELD=IGNORE

  REASON: An invalid column was found.

  SYSTEM ACTION: Processing continues as INVFIELD=IGNORE was specified.

  USER ACTION: None.

- THIS TABLE CONTAINS INVALID COLUMN xxxxxxxxxxxxx

  REASON: The column xxxxxxxxxxxxx was an invalid column name for the type of table.

  SYSTEM ACTION: Processing may continue, or the request is aborted. Action depends upon the INVFIELD= parameter.

  USER ACTION: None if processing continues. If request is aborted, correct and resubmit.

- NO SUMMARIZED OUTPUT RECORDS ARE PRODUCED

  REASON: No summarized rows are produced.

  SYSTEM ACTION: Processing continues.

  USER ACTION: None.

## Error Messages

The following error messages are produced by the table load DPSUMLD program. All messages are written to the DPSYSOUT data set.

- CANNOT OPEN DPLOAD DATASET

  REASON: DPLOAD DD was not defined.

  SYSTEM ACTION: Load request aborted.

  USER ACTION: Add DPLOAD DD for LOAD requests.

- KEYWORD xxxxxxxx MUST BE SPECIFIED WITH VALUE

  REASON: The keyword xxxxxxxx does not have a default value and a value must be specified.

  SYSTEM ACTION: The request is not processed.

  USER ACTION: Supply correct value.

- TABLE OR SUMMKEY MUST BE SPECIFIED

  REASON: Either TABLE or SUMMKEY must be specified in a summarization request.

  SYSTEM ACTION: Request not processed.

  USER ACTION: Supply correct value.

- EITHER FROMTBL OR FROMDD MUST BE SPECIFIED, BUT NOT BOTH

  REASON: FROMTBL and FROMDD are specified. One or the other must be specified, but not both.

  SYSTEM ACTION: Request not processed.

  USER ACTION: Supply correct value.

- EITHER EXPDT OR REPTD MUST BE SPECIFIED, BUT NOT BOTH

  REASON: EXPDT and REPTD are specified. One or the other must be specified, but not both.

  SYSTEM ACTION: Request not processed.

  USER ACTION: Supply correct value.

- SUMMKEY - LUWIDNID - IS ALLOWED ONLY IF TYPE IS SUMACCT OR SUMDDFACCT

  REASON: The SUMMKEY - LUWIDNID - is allowed only if the type is SUMACCT or SUMDDFACCT.

  SYSTEM ACTION: Request not processed.

  USER ACTION: Do not use LUWIDNID as the SUMMKEY. Correct and resubmit.

- SUMMKEY - LUWIDLUNM - IS ALLOWED ONLY IF TYPE IS SUMACCT OR SUMDDFACCT

  REASON: The SUMMKEY - LUWIDLUNM - is allowed only if the type is SUMACCT or SUMDDFACCT.

  SYSTEM ACTION: Request not processed.

  USER ACTION: Do not use LUWIDLUNM as the SUMMKEY. Correct and resubmit.

- SUMMKEY - DDFLOCATION - IS ALLOWED ONLY IF TYPE IS DDFACCT OR SUMDDFACCT

  REASON: The SUMMKEY - DDFLOCATION - is allowed only if the type is DDFACCT or SUMDDFACCT.

  SYSTEM ACTION: Request not processed.

  USER ACTION: Do not use DDFLOCATION as the SUMMKEY. Correct and resubmit.

- ***xxxxxxxxxxxxxx IS NOT A VALID SUMMKEY VALUE

  REASON: The column xxxxxxxxxxxxxx is not a valid SUMMKEY column.

  SYSTEM ACTION: Request not processed.

  USER ACTION: Supply correct value.

- DB2 ERROR MESSAGES RECEIVED

  REASON: DB2 has issued error messages.

  SYSTEM ACTION: Request not processed.

  USER ACTION: Refer to the DB2 error message received.

- PROCESSING ABORTED AS INVFIELD=ABORT (DEFAULT)

  REASON: An invalid column was found.

  SYSTEM ACTION: Request not processed.

  USER ACTION: Specify INVFIELD=IGNORE or redefine the table without the invalid column.

- DATA AREA RUNS OUT. CONTACT BMC SOFTWARE

  REASON: The internal data area for SQLDA runs out.

  SYSTEM ACTION: Request not processed.

  USER ACTION: Contact BMC Software.

- CANNOT OPEN xxxxxxxx DD, PROCESS TERMINATED

  REASON: The DD xxxxxxxx statement cannot be opened.

  SYSTEM ACTION: Request not processed.

  USER ACTION: Correct the DD statement and resubmit.

- SORT HAS ENCOUNTERED ERROR. SUMMARY PROCESS TERMINATED

  REASON: Errors have occurred in the sort.

  SYSTEM ACTION: Request not processed.

  USER ACTION: Determine the reason for the sort error from the sort messages.

- COMBINE FIELD - xxxxxxxxxxxxx IS AN INVALID FIELD, COMBINE ABORTED

  REASON: The field xxxxxxxxxxxxx defined in the COMBINE parameter is an invalid field.

  SYSTEM ACTION: COMBINE aborted.

  USER ACTION: Supply correct values for the COMBINE parameter.

- COMBINE FIELD - xxxxxxxxxxxxxx HAS INVALID ATTRIBUTE, COMBINE ABORTED

  REASON: The COMBINE xxxxxxxxxxxxxx field has an invalid or incompatible attribute.

  SYSTEM ACTION: COMBINE aborted.

  USER ACTION: Supply correct values for the COMBINE parameter.

- SUMMKEY - xxxxxxxxxxxxx IS NOT DEFINED IN RECEIVING TABLE

  REASON: The column xxxxxxxxxxxxx defined in SUMMKEY is not defined in the receiving table.

  SYSTEM ACTION: Request not processed.

  USER ACTION: Supply correct values for SUMMKEY and/or TABLE.

- DB2 CONNECTION NOT ESTABLISHED

  REASON: DB2 connection cannot be established.

  SYSTEM ACTION: No further request can be accepted.

  USER ACTION: Refer to subsequent messages for diagnostics.

- DB2 CONNECTION MUST BE ESTABLISHED BEFORE PROCESSING ANY REQUEST

  REASON: A request cannot be processed until the DB2 connection is established.

  SYSTEM ACTION: Request not processed.

  USER ACTION: Find the reason for the DB2 connection failure.

- MAKE SURE GLOBAL STATEMENT IS CODED

  REASON: DB2 connection is not established.

  SYSTEM ACTION: No further requests can be accepted.

  USER ACTION: Assure GLOBAL statement was coded; if it was, continue looking for the reason for the DB2 connection failure.

- JXR3010C - INTERNAL ERROR INVALID ROUTINE CODE - code

  REASON: Internal processing error. The parameter is invalid.

  SYSTEM ACTION: Processing ends.

  USER ACTION: Contact BMC Software.

- JXR3011E - SYNTAX ERROR - error

REASON:A syntax error was found in the input.

where

error = UNMATCHED PARENTHESIS
NO CONTINUATION FOUND
IMBEDDED LEFT PARENTHESIS
EXTRA RIGHT PARENTHESIS
NO ENDING QUOTE
INVALID QUOTED STRING
ODD NUMBER OF HEX DIGITS
INVALID HEX DIGITS

The data is considered in error and is not used in the processing.

SYSTEM ACTION: Processing ends.

USER ACTION: Correct the syntax error and resubmit.

- JXR3012E - DUPLICATE KEY ENTERED - key

REASON: The keyword *key* was typed twice within one control statement. The second value is ignored.

SYSTEM ACTION: Statement is discarded.

USER ACTION: Correct and resubmit.

- JXR3013E - INVALID KEYWORD key

REASON: The keyword *key* is not valid in this control statement.

SYSTEM ACTION: Statement is discarded.

USER ACTION: Correct and resubmit.

- JXR3014I - VALUE ‘value’ FOR INVALID KEY SKIPPED

REASON: An invalid or duplicate keyword was detected which had an associated value. This value is skipped.

SYSTEM ACTION: Statement is discarded.

USER ACTION: Correct and resubmit.

- JXR3015E - DATA ERROR FOR key - error desc.

  REASON: The data entered for a keyword failed an edit. The error description specifies which edit was failed. The possible values include

  - MISSING 2ND RANGE VALUE
  - MUST HAVE A VALUE
  - date edit messages
  - EXCEEDED NUMBER OF VALUES
  - INPUT DATA TOO LONG
  - INPUT DATA TOO SHORT
  - INVALID DATA FOR INPUT
  - INPUT IS NOT NUMERIC
  - VALUE NOT ALLOWED
  - VALUE REQUIRED
  - INVALID PARENTHESES

  SYSTEM ACTION: Statement is discarded.

  USER ACTION: Correct and resubmit.

- JXR3016W - NULL KEY VALUE FOUND

  REASON: A null value was found in scanning for next keyword. This is usually from two adjacent commas.

  SYSTEM ACTION: Skip to the next keyword.

  USER ACTION: None.

# Chapter 3. Data Summary and Purge Process (DPRSUM)

This chapter describes the DPRSUM batch job. The purpose of this procedure is to

- Summarize rows in the DB2 detail accounting table or package accounting table into a summary table to save space and reduce processing costs
- Summarize rows in a summary accounting table into a table that is summarized over a longer interval
- Purge outdated data from the performance data tables

**Note:** Multiple summarize and purge processes can be specified in one run.

The summarization process accesses a detail accounting table and produces a new or updated summary accounting table. The purge process accesses any performance data table and deletes rows with a DATETIME earlier than the specified purge date or retention period.

## Summarization Strategy Considerations

Flexibility is built into the summarization process to allow for differences in installation requirements. Summarization should be performed, however, because the individual accounting records require considerable storage space.

It is very important to plan your summarization strategy prior to implementing the summarization process. For example, either changing the summarization keys or intervals, or purging from the detail table for a partial summary period can produce inconsistent reporting results.

There are several possible summarization strategies. For example:

- For greatest retention of detail, summarize the accounting data by location, subsystem, plan, authorization ID, connection type, and correlation ID. This strategy provides the greatest level of detail, but requires considerable storage. It does maintain uniqueness across workload types (such as batch or TSO) and CICS transaction code or IMS PSBNAME.
- For less detailed data, summarize by location, subsystem, and plan. If the information is summarized any further, individual identity is lost and the summary begins to resemble the statistics records. The distributed summary reports by AUTHID then become unusable.
- To compromise between storage economy and usable detail, summarize by location, subsystem, plan, and authorization ID. This strategy retains accountability to the user (authorization ID) for a controlled period of time.
- After determining the key and estimating the number of unique occurrences to be expected with this specification, decide on one or more reasonable summarization intervals based on your reporting needs and available storage. A first-level summary interval can be as low as 5 or 30 minutes for greater detail, 60 minutes to allow for hourly reporting, or as long as a full day. This table can be summarized further into a second summary table.

**Note:** The default specifications are to load summary tables with intervals of 30 minutes in the SMF extract job (DPRSMF) and to summarize weekly to a second set of tables with intervals of 1 day in this job (DPRSUM).

To improve results:

- To match summary totals against the statistics reports, define the summary interval the same as the DB2 statistics interval.
- For ease in reading the resulting reports, define the summary start time on an even time period. For example, specify 05:00:00 to start summarization at 5:00am.
- To make it possible to begin a second table at weekly or monthly intervals, specify the receiving table name during the summarization process.
- To free storage space, purge outdated or summarized detail data. The detail rows should be purged after summarization. How long the summary data should be kept depends on its use.

## DPRSUM Job Control Statements

Summarization and data purge of the performance data tables is a single-step procedure, as shown in Figure 5 on page 53.

Following is a description of the job control language (JCL) statements used in Figure 5 on page 53 in "DPSUMLD Control Statements" on pages 27 through 35.

## Summary/Purge Procedure

STEP 3
Specifies the name of the Performance Reporter summarize and load program as:

PGM=DPSUMLD

STEPLIB DD
Defines the DB2 DSNLOAD data set and Performance Reporter load library.

DPSYSOUT DD
Defines the data set for DPSUMLD results.

SYSOUT DD
Defines the output class for sort messages.

SYSUDUMP DD
Defines the dump data set for problem determination.

SORTWKnn DD
Defines work data sets for data sorting; nn is a numeric.

SUMMACCT DD
Defines the output data set for the SUMMARIZE process. This file is passed to Step 4 SUMMACCT.

DDFSACCT
Defines the output data set for the DDF SUMMARIZE process. This file is passed to Step 4 DDFSACCT.

PKGSACCT
Defines the output data set for the package SUMMARIZE process. This file is passed to Step 4 PKGSACCT.

BUFSACCT
Defines the output data set for the buffer pool SUMMARIZE process. This file is passed to Step 4 BUFSACCT.

DPLOAD DD
Defines the data set for DB2 LOAD control statements.

DPSYSIN
Refer to "DPSUMLD Control Statements" on pages 27 through 35 for information on control statements.

## Load into DB2 Tables

STEP 4
Specifies the name of the DB2 Utility load program as

PGM=DSNUTILB

and the region required to run the program.

Steps to load DB2 tables use the DB2 Load Utility, DSNUTILB. You must modify your JCL if your installation uses a different utility. The utility must be able to interpret LOAD control statements using the same format that DSNUTILB uses.

STEPLIB DD
Defines the program library containing the DB2 Utility load program.

SYSPRINT
Defines the data set for program messages.

UTPRINT DD
Defines the data set for DFSORT messages.

SYSUDUMP DD
Defines the dump data set for problem determination.

SORTOUT DD
Defines the data set for the SORT output.

SORTWKnn DD
Defines work data sets for data sorting; nn is a numeric.

SUMMACCT
Defines the input data set for the ACCT SUMMARIZE LOAD utility.

DDFSACCT
Defines the input data set for the DDFACCT SUMMARIZE LOAD utility.

PKGSACCT
Defines the input data set for the PKGACCT SUMMARIZE LOAD utility.

BUFSACCT
Defines the input data set for the BUFACCT SUMMARIZE LOAD utility.

SYSIN DD
Defines the input data set containing the DB2 LOAD utility control statement.

Figure 5 shows sample job control statements for the data summary and purge process.

```
//DPRSUM   JOB
//*------------------------------------------------------------*
//*       SUMMARIZE OR RE-SUMMARIZE ACCOUNTING TABLES.         *
//*                                                            *
//*       ===> DETERMINE SUMMARIZATION INTERVAL AND FREQUENCY  *
//*            DEFAULTS ARE: 1 DAY INTERVAL, RUN WEEKLY        *
//*                                                            *
//*       ===> ADJUST CONTROL STATEMENTS:                      *
//*            RELATIVE START DATE OF SUMMARIZATION (SUMMSTART)*
//*            SUMMARIZATION INTERVAL (SUMMINT)                *
//*            PURGE RETENTION PERIOD (RETPD) (DEFAULT 60 DAYS)*
//*------------------------------------------------------------*
//DPPSUM  PROC HIDP='HILVL.RUN.LIB',              <-- UPDATE
//             HIDB2=DSN510,                      <-- VERIFY
//             UNIT=SYSDA,                        <-- VERIFY
//*                     THE FOLLOWINGS ARE FOR DB2 LOAD UTILITY
//             SYSTEM=DDDD,                       <-- UPDATE
//             UID=UTIDSUM                        <-- UPDATE
//*------------------------------------------------------------*
//*            BUILD DB2 LOAD CONTROL STATEMENTS,              *
//*            SUMMARIZE ACCOUNTING AND DDF ACCOUNTING RECORDS,*
//*            PURGE OLD RECORDS                               *
//*------------------------------------------------------------*
//*
//STEP3   EXEC PGM=DPSUMLD,REGION=4096K
//STEPLIB   DD DSN=&HIDB2..SDSNLOAD,DISP=SHR
//          DD DISP=SHR,DSN=&HIDP..BBLINK
//DPSYSOUT  DD SYSOUT=*            PROGRAM MESSAGES
//SYSOUT    DD SYSOUT=*            SORT MESSAGE
//SORTMSG   DD SYSOUT=*            SORT MESSAGE
//SYSUDUMP  DD SYSOUT=*
//SORTWK01  DD UNIT=&UNIT,SPACE=(CYL,(100,50))
//SORTWK02  DD UNIT=&UNIT,SPACE=(CYL,(100,50))
//SORTWK03  DD UNIT=&UNIT,SPACE=(CYL,(100,50))
//SORTWK04  DD UNIT=&UNIT,SPACE=(CYL,(100,50))
//SORTWK05  DD UNIT=&UNIT,SPACE=(CYL,(100,50))
//SORTWK06  DD UNIT=&UNIT,SPACE=(CYL,(100,50))
//DPSYSIN   DD DUMMY             CONTROL STATEMENTS
//DPLOAD    DD UNIT=&UNIT,DISP=(,PASS),SPACE=(TRK,(1,1),RLSE),
//             DCB=(RECFM=FB,LRECL=80,BLKSIZE=3200)
//*                      USER-DEFINED DD STATEMENTS
//SUMMACCT  DD UNIT=&UNIT,DISP=(,PASS),SPACE=(CYL,(300,50),RLSE)
//             DCB=(LRECL=2048,BLKSIZE=23760,RECFM=VB)
//DDFSACCT  DD UNIT=&UNIT,DISP=(,PASS),SPACE=(CYL,(10,2),RLSE),
//             DCB=(LRECL=2048,BLKSIZE=23760,RECFM=VB)
//PKGSACCT  DD UNIT=&UNIT,DISP=(,PASS),SPACE=(CYL,(10,2),RLSE),
//             DCB=(LRECL=2048,BLKSIZE=23760,RECFM=VB)
//BUFSACCT  DD UNIT=&UNIT,DISP=(,PASS),SPACE=(CYL,(10,2),RLSE),
//             DCB=(LRECL=2048,BLKSIZE=23760,RECFM=VB)
```

Figure 5.  Sample DPRSUM Job Control Statements (Page 1 of 3)

```
//*------------------------------------------------------------*
//*            LOAD INTO DB2 TABLES                            *
//*------------------------------------------------------------*
//*
//STEP4   EXEC PGM=DSNUTILB,REGION=4096K,
//             PARM='&SYSTEM,&UID'
//STEPLIB   DD DSN=&HIDB2..SDSNLOAD,DISP=SHR
//SYSPRINT  DD SYSOUT=*
//SORTMSG   DD SYSOUT=*
//UTPRINT   DD SYSOUT=*
//SYSUDUMP  DD SYSOUT=*
//SORTOUT   DD UNIT=&UNIT,SPACE=(CYL,(10,2))
//SORTWK01  DD UNIT=&UNIT,SPACE=(CYL,(5,2))
//SORTWK02  DD UNIT=&UNIT,SPACE=(CYL,(5,2))
//SORTWK03  DD UNIT=&UNIT,SPACE=(CYL,(5,2))
//SYSUT1    DD UNIT=&UNIT,SPACE=(CYL,(5,2))
//SYSIN     DD DISP=(OLD,DELETE),DSN=*.STEP3.DPLOAD
//*
//SUMMACCT  DD DISP=(OLD,DELETE),DSN=*.STEP3.SUMMACCT
//DDFSACCT  DD DISP=(OLD,DELETE),DSN=*.STEP3.DDFSACCT
//PKGSACCT  DD DISP=(OLD,DELETE),DSN=*.STEP3.PKGSACCT
//BUFSACCT  DD DISP=(OLD,DELETE),DSN=*.STEP3.BUFSACCT
//        PEND
//*
//JS010   EXEC DPPSUM
//STEP3.DPSYSIN DD *
   GLOBAL LOG=NO,SSID=DDDD,INVFIELD=IGNORE,PLAN=DPSUMLD    <-- UPDATE
*
*      --- SUMMARIZE LAST 7 DAYS INTO DAILY TOTALS ---
*
   SUMMARIZE TYPE=SUMACCT,FROMTBL=DMRPR.DMRACSUM,
       SUMMSTART=*_7,
       SUMMEND=*,
       SUMMINT=1D,
       SUMMKEY=(LOCATION,SUBSYSTEM,PLANNAME,AUTHID),
       TABLE=DMRPR.DMRACSM2,TODD=SUMMACCT
   LOAD   TYPE=SUMACCT,TABLE=DMRPR.DMRACSM2,FROMDD=SUMMACCT
*
*      --- SUMMARIZE LAST 7 DAYS DDF INTO DAILY TOTALS ---
*
   SUMMARIZE TYPE=SUMDDFACCT,FROMTBL=DMRPR.DMRADSUM,
       SUMMSTART=*_7,
       SUMMEND=*,
       SUMMINT=1D,
       SUMMKEY=(LOCATION,SUBSYSTEM,PLANNAME,AUTHID,DDFLOCATION),
       TABLE=DMRPR.DMRADSM2,TODD=DDFSACCT
   LOAD   TYPE=SUMDDFACCT,TABLE=DMRPR.DMRADSM2,FROMDD=DDFSACCT
*
*      --- SUMMARIZE LAST 7 DAYS PKG INTO DAILY TOTALS ---
*
   SUMMARIZE TYPE=SUMPKGACCT,FROMTBL=DMRPR.DMRAPSUM,
       SUMMSTART=*_7,
       SUMMEND=*,
       SUMMINT=1D,
       SUMMKEY=(LOCATION,SUBSYSTEM,PLANNAME,AUTHID,PROGRAMNAME),
       TABLE=DMRPR.DMRAPSM2,TODD=PKGSACCT
   LOAD   TYPE=SUMPKGACCT,TABLE=DMRPR.DMRAPSM2,FROMDD=PKGSACCT
```

Figure 5.  Sample DPRSUM Job Control Statements (Page 2 of 3)

```
*      --- SUMMARIZE LAST 7 DAYS BUF INTO DAILY TOTALS ---
*
   SUMMARIZE TYPE=SUMBUFACCT,FROMTBL=DMRPR.DMRABSUM,
       SUMMSTART=*_7,
       SUMMEND=*,
       SUMMINT=1D,
       SUMMKEY=(LOCATION,SUBSYSTEM,PLANNAME,AUTHID,BPNAME),
       TABLE=DMRPR.DMRABSM2,TODD=BUFSACCT
   LOAD   TYPE=SUMBUFACCT,TABLE=DMRPR.DMRABSM2,FROMDD=BUFSACCT
*
   PURGE  TYPE=SUMACCT,TABLE=DMRPR.DMRACSM2,RETPD=60D
   PURGE  TYPE=SUMDDFACCT,TABLE=DMRPR.DMRADSM2,RETPD=60D
   PURGE  TYPE=SUMPKGACCT,TABLE=DMRPR.DMRAPSM2,RETPD=60D
   PURGE  TYPE=SUMBUFACCT,TABLE=DMRPR.DMRABSM2,RETPD=60D
//
```

Figure 5. Sample DPRSUM Job Control Statements (Page 3 of 3)

## Summarization/Purge Return Codes and Error Messages

The return codes and messages produced by the summary/purge program are the same as those produced by the table load program. See “DPRSMF Return Codes and Error Messages” on page 36.

# Chapter 4. SMF Reporting Facilities (DPREPORT)

This chapter describes the DPREPORT batch job. This job produces accounting or statistics reports, in either a short or long format. The input for this job is an accounting or statistics extract file created during the DPRSMF SMF extract job.

The DPRSMF job stream can be tailored to produce accounting and statistics reports during the daily SMF extract, either in addition to, or in place of, loading the data into DB2 tables. The extract files can be saved for later reporting with the DPRSMFR job.

Both DPRSMF and DPRSMFR contain two steps, each of which invokes the DPREPORT program:

- Step 5 produces a statistics report. The default is a statistics long report for the total interval included in the input data. The control statements can be modified to select data by date, time, relative day, SMFID, or DB2 ID. A different summarization interval can also be specified.
- Step 6 produces an accounting report. The default is an accounting long report by plan for the total interval included in the input data. The control statements can be modified to select data by date, time, relative day, SMFID, DB2 ID, or various thread identifiers. The summarization keys and interval can also be modified.

## JCL Control Statements

The following list describes the job control language (JCL) statements required to execute the DPREPORT program to produce either an accounting or a statistics report:

STEP 5 (statistics) / STEP 6 (accounting)

Specifies the name of the SMF report program as

PGM=DPREPORT

and the region required to run the program.

STEPLIB DD

Defines the program library containing the DPREPORT load module.

DPDSTAT DD

Contains the statistics records created from the DB2 SMF 100 statistics records. This file is passed or kept from step 2 DPDSTAT in DPRSMF. This statement is required if REPORT=STAT is coded in SYSIN (Step 5).

DPDACCT DD

Contains the accounting records created from the DB2 SMF 101 accounting records. This file is passed or kept from step 1 DPDACCT in DPRSMF. This statement is required if REPORT=ACCT is coded in SYSIN (Step 6).

**Note:** It is also possible to keep and use here the 1-3 output summary accounting files produced in step 3 of DPRSMF if granularity at the thread level is not needed for reporting. This would reduce the volume of data saved.

SYSPRINT DD

Defines the data set for program messages.

DPDPRINT DD

Defines the data set for accounting or statistics reports.

SYSOUT DD

Defines the output class for sort messages.

SYSUDUMP DD

Defines the dump data set for problem determination.

SORTWKnn DD

Defines work data sets for data sorting; nn is a numeric.

SYSIN DD

**Note:** Control statements are keyword format, one statement per line. The keyword must start in column 1.

The following control statements are valid:

REPORT=STAT|ACCT

Required. For a statistics report, specify REPORT=STAT. For an accounting report, specify REPORT=ACCT.

REPTYP=SHORT|LONG

Required. Defines the format of the report.

RLOCID=llllllllllllllll

Optional. Selects records from a single location or a generic group of locations (l*). The default is to select all records.

RDB2ID=dddd

Optional. Selects records from DB2 subsystem dddd or a generic group of DB2s (d*). The default is to select all records.

RGROUP=gggggggg

Optional. Selects records from a DB2 data sharing group or a generic group of DB2 data sharing groups (g*). The default is to select all records.

RMEMBR=mmmmmmmm

Optional. Selects members of a DB2 data sharing group or a generic group of members (m*). The default is to select all members.

RSCOPE=MEMBER|GROUP

Optional. MEMBER shows the activity of each data sharing group member. GROUP shows summarized activity for all members within the DB2 data sharing group. MEMBER is the default.

DPRDAY=0, or -1 to -7

Optional. Selects records by relative day, specified as a decimal number from -1 to -7 that is subtracted from the current date. Specify zero to select the current date. The report interval will be 00:00:00 to 23:59:59 hours for the computed date.

**Note:** FRDATE and TODATE are ignored when DPRDAY is specified; however, FRTIME and TOTIME are supported for the single date selected by DPRDAY.

DLYTIM=N|Y

Optional. Y applies FRTIME and TOTIME to each day within FRDATE and TODATE to allow processing of only selected times for several days (single shift). The default of N results in FRTIME being applied to FRDATE only and TOTIME applied to TODATE only and all records are processed for days between FRDATE and TODATE.

FRDATE=yyyymmdd

Optional. Selects records starting from this date.

FRTIME=hhmmss

Optional. Selects records starting from this time.

TODATE=yyyymmdd

Optional. Selects records ending at this date.

TOTIME=hhmmss

Optional. Selects records ending at this time.

**Note:** An hhmmss value of 240000 is invalid. Use 235959 instead.

INTVAL=nnnn

Optional. Defines a reporting interval in minutes. The default is one interval for all the input data, or for the total reporting interval defined by FRDATE/FRTIME to TODATE/TOTIME. If INTVAL is specified, the FROM/TO date/time keywords or DPRDAY are also required.

SORTNM=nnnnnnnn

Optional. Defines the name of a different sort program than 'SORT'.

RORDER=(key1,key2)

Optional, and valid only for REPORT=ACCT. Defines the summarization and reporting keys for an accounting report. The following keywords are supported:

- CONNID - Connection Name
- CORRID - Correlation ID
- ORAUTH - Original Authorization ID
- PLANAM - Plan Name
- PRAUTH - Primary Authorization ID
- REQLOC - DDF Location (local)
- CONTYP - Connection Type
- PAKAGE - First package/DBRM executed (accounting class 7 required)

If RORDER is not specified, summarization is done only by time interval. Examples are

```
RORDER=(PRAUTH,PLANAM)   To summarize by authid and plan
RORDER=(CONNID)          To summarize by connection ID
```

RSMFID=ssss

Optional. Selects records from a single SMF system ID (ssss) or a generic group of system IDs (s*). The default is to select all records.

DPTHRD=N|Y

Optional, and valid only for REPORT=ACCT. Y specifies that one accounting report should be printed per thread. For example, you may want to use this to analyze large batch programs. Do not use Y with a large amount of input data. Use of filters is recommended. The default is N. When DPTHRD=Y, RORDER is invalid.

FILTYP=(key1,key2)

Optional, and valid only for REPORT=ACCT. Defines the thread identifiers used for selection criteria. The following keywords are supported:

- CONNID - Connection Name
- CORRID - Correlation ID
- ORAUTH - Original Authorization ID
- PLANAM - Plan Name
- PRAUTH - Primary Authorization ID
- REQLOC - DDF Location (local)
- CONTYP - Connection Type
- PAKAGE - First package/DBRM executed (accounting class 7 required)

Key1 defines which identifier in the input accounting records is matched to the selection criteria specified in FILTR1. Key2 defines which identifier is matched to the selection criteria specified in FILTR2. If FILTYP is not specified, all input records within the specified time interval are selected. For example:

```
FILTYP=(CONNID,CORRID)   FILTR1 and FILTR2 must be coded
FILTYP=(PLANAM)          FILTR1 must be coded
```

FILTR1=(id1,id2,....)

Optional, and valid only for REPORT=ACCT. Required if FILTYP is specified. Defines a list of matching values (generic with * allowed) for the first keyword of FILTYP.

FILTR2=(id1,id2,....)

Optional, and valid only for REPORT=ACCT. Required if a second key is specified in FILTYP. Defines a list of matching values (generic with * allowed) for the second keyword of FILTYP.

**Note:** FILTR1 and FILTR2 can define up to 64 identifiers separated by commas. Identifiers must not be split at the end of the statement. Continuation cards must begin with at least one blank.

Examples are

```
 FILTYP=(CONNID)
 FILTR1=(BATCH,TSO)
```

or

```
 FILTYP=(CONNID,CORRID)
 FILTR1=(BATCH,TSO)
 FILTR2=(PSR1A001,PSR1A002,YTQ*)
```

**Note:** Parentheses must be specified with the parameters for RORDER, FILTYP, FILTR1 and FILTR2, even when only one value is specified.

BUFDTL=N|Y

Optional, and valid only if REPORT=STAT. Y requests buffer pool and global buffer pool reporting of individual buffer pools. The default is N and shows only buffer pool summaries of all 4K and 32K buffer pools and global buffer pools.

## Sample DPREPORT Job Control Statements for Statistics

Figure 6 shows sample job control statements for producing statistics reports.

```
//DPRSTAT JOB
//*-------------------------------------------------------------
//*         JOB TO PRODUCE A STATISTICS SHORT REPORT           -
//*         FOR OCT. 16, 10.00.00 TO 17.59.59 HOURS,           -
//*                      (10 a.m. TO 6 P.M.)                   -
//*         SUMMARIZED IN INTERVALS OF 2 HOURS.                -
//*-------------------------------------------------------------
//STEP1    EXEC PGM=DPREPORT
//STEPLIB  DD   DSN=HILVL.DPRLOAD,DISP=SHR
//DPDSTAT  DD   DSN=HILVL.STAT31,DISP=SHR
//SYSPRINT DD   SYSOUT=*
//DPDPRINT DD   SYSOUT=*
//SYSOUT   DD   SYSOUT=*
//SYSUDUMP DD   SYSOUT=*
//SORTWK01 DD   UNIT=SYSDA,SPACE=(TRK,(45,15)),DISP=(,DELETE,DELETE)
//SORTWK02 DD   UNIT=SYSDA,SPACE=(TRK,(45,15)),DISP=(,DELETE,DELETE)
//SORTWK03 DD   UNIT=SYSDA,SPACE=(TRK,(45,15)),DISP=(,DELETE,DELETE)
//SORTWK04 DD   UNIT=SYSDA,SPACE=(TRK,(45,15)),DISP=(,DELETE,DELETE)
//SORTWK05 DD   UNIT=SYSDA,SPACE=(TRK,(45,15)),DISP=(,DELETE,DELETE)
//SORTWK06 DD   UNIT=SYSDA,SPACE=(TRK,(45,15)),DISP=(,DELETE,DELETE)
//SORTWK07 DD   UNIT=SYSDA,SPACE=(TRK,(45,15)),DISP=(,DELETE,DELETE)
//SORTWK08 DD   UNIT=SYSDA,SPACE=(TRK,(45,15)),DISP=(,DELETE,DELETE)
//SYSIN    DD   *
REPORT=STAT
REPTYP=SHORT
RSMFID=SYSB
RDB2ID=DB2F
FRDATE=19951016
FRTIME=100000
TODATE=19951016
TOTIME=175959
INTVAL=120
//
```

Figure 6. Sample DPREPORT Job Control Statements for Statistics Reports

## Sample DPREPORT Job Control Statements for Accounting

Figure 7 shows sample job control statements for producing accounting reports.

```
//DPRACCT JOB
//*-------------------------------------------------------------
//*         JOB TO PRODUCE AN ACCOUNTING LONG REPORT            -
//*         ORDERED BY PRIMARY AUTHID AND PLAN NAME             -
//*         FROM 00:00:00 - 23:59:59 HOURS FOR OCT. 16 - 18,    -
//*         SUMMARIZED IN INTERVALS OF 8 HOURS (480 MINUTES).   -
//*         ONLY AUTHIDS OF BPL2X OR THOSE BEGINNING WITH CJN*  -
//*         ARE SELECTED FOR THIS REPORT.                       -
//*-------------------------------------------------------------
//STEP1    EXEC PGM=DPREPORT,REGION=4096K
//STEPLIB  DD   DSN=HILVL.DPRLOAD,DISP=SHR
//DPDACCT  DD   DSN=HILVL.ACCT31,DISP=SHR
//SYSPRINT DD   SYSOUT=*
//DPDPRINT DD   SYSOUT=*
//SYSOUT   DD   SYSOUT=*
//SYSUDUMP DD   SYSOUT=*
//SORTWK01 DD   UNIT=SYSDA,SPACE=(TRK,(45,15)),DISP=(,DELETE,DELETE)
//SORTWK02 DD   UNIT=SYSDA,SPACE=(TRK,(45,15)),DISP=(,DELETE,DELETE)
//SORTWK03 DD   UNIT=SYSDA,SPACE=(TRK,(45,15)),DISP=(,DELETE,DELETE)
//SORTWK04 DD   UNIT=SYSDA,SPACE=(TRK,(45,15)),DISP=(,DELETE,DELETE)
//SORTWK05 DD   UNIT=SYSDA,SPACE=(TRK,(45,15)),DISP=(,DELETE,DELETE)
//SORTWK06 DD   UNIT=SYSDA,SPACE=(TRK,(45,15)),DISP=(,DELETE,DELETE)
//SORTWK07 DD   UNIT=SYSDA,SPACE=(TRK,(45,15)),DISP=(,DELETE,DELETE)
//SORTWK08 DD   UNIT=SYSDA,SPACE=(TRK,(45,15)),DISP=(,DELETE,DELETE)
//SYSIN    DD   *
REPORT=ACCT
REPTYP=LONG
RSMFID=SYSB
RDB2ID=DB2F
FRDATE=20010516
FRTIME=000000
TODATE=20010518
TOTIME=240000
INTVAL=480
RORDER=(PRAUTH,PLANAM)
FILTYP=(PRAUTH)
FILTR1=(BPL2X,CJN*)
//
```

Figure 7. Sample DPREPORT Job Control Statements for Accounting Reports

## DPREPORT Messages

The following messages are produced by the DPREPORT program:

- DPRM003A mmmmmmmm LOAD FAILED

  REASON: DPREPORT initialization failed to load the module mmmmmmmm.

  SYSTEM ACTION: DPREPORT terminates with a return code 12.

  USER ACTION: Check the load library specification in the JCL. Correct and resubmit.

- DPRM004A GETMAIN FAILED

  REASON: DPREPORT failed to obtain one of its work areas.

  SYSTEM ACTION: DPREPORT terminates with a return code 12.

  USER ACTION: Verify control statements to ensure that INTVAL is not specified without FRDATE/TODATE statements. Otherwise, increase the region size.

- DPRM005A dddddddd OPEN ERROR

  REASON: DPREPORT failed to open the data set referenced by ddname dddddddd.

  SYSTEM ACTION: DPREPORT terminates with a return code 12.

  USER ACTION: Correct JCL and resubmit.

- DPRM006A mmmmmmmm LINK FAILED

  REASON: DPREPORT failed to link to the sort program whose name is mmmmmmmm.

  SYSTEM ACTION: DPREPORT terminates with a return code 12.

  USER ACTION: Determine reason for failure. Correct and resubmit.

- DPRM007A kkkkkk INVALID RORDER/FILTER KEYWORD

  REASON: An invalid keyword kkkkkk has been coded in an RORDER or FILTYP statement.

  SYSTEM ACTION: DPREPORT terminates with a return code 12.

  USER ACTION: Correct and resubmit.

- DPRM008A pppppp INVALID OR OMITTED

  REASON: The statement pppppp has been coded incorrectly, or is required and has been omitted.

  SYSTEM ACTION: DPREPORT terminates with a return code 12.

  USER ACTION: Correct or add statement.

- DPRM009A dddddddd MISSING OR INVALID

  REASON: The DD statement for ddname ddddddd is missing or invalid.

  SYSTEM ACTION: DPREPORT terminates with a return code 12.

  USER ACTION: Correct JCL and resubmit.

- DPRM010A INVALID CURRENT DATE (<1980)

  REASON: yyyy of statement FRDATE and TODATE must be > 1980.

  SYSTEM ACTION: DPREPORT terminates with a return code 12.

  USER ACTION: Correct and resubmit.

- DPRM011A DPTHRD INCOMPATIBLE WITH STAT OR RORDER

  REASON: The DPTHRD statement is valid only for REPORT=ACCT, not STAT. It is incompatible with the RORDER statement.

  SYSTEM ACTION: DPREPORT terminates with a return code 12.

  USER ACTION: Correct statement and resubmit.

- DPRM012A CONVERT MINUTES TO DATE INTERNAL ERROR OR INTVAL CONVTOD FAILURE

  REASON: Internal error in the date or time conversion routine.

  SYSTEM ACTION: DPREPORT terminates with a return code 12.

  USER ACTION: Contact BMC Software.

- DPRA011A tttttttttttttttttttttttt INVALID BLKID

  REASON: Internal error in the accounting report definitions. For a time failure (CONVTOD), check to see if you have used the invalid value of 240000.  If so, use 235959 instead.

  SYSTEM ACTION: DPREPORT terminates with a return code 12

  USER ACTION: Contact BMC Software.

- DPRA012A NO RECORD MATCHING SELECTION CRITERIA:

  REASON: No input accounting records were found that match the criteria specified.

  SYSTEM ACTION: DPREPORT terminates with a return code 16.

  USER ACTION: Determine error in selection criteria or input file.

- DPRA013A INTERNAL ERROR: INVALID IVC FLAG BYTE

  REASON: Internal error.

  SYSTEM ACTION: DPREPORT terminates with a return code 12.

  USER ACTION: Contact BMC Software.

- DPRA014A INVALID VARIABLE DESCRIPTOR BLOCK

  REASON: Internal error in the accounting DDF or package report definitions.

SYSTEM ACTION: DPREPORT terminates with a return code 12.

USER ACTION: Contact BMC Software.

- DPRS011A tttttttttttttttttttttttt INVALID BLKID

  REASON: Internal error in the statistics report definitions.

  SYSTEM ACTION: DPREPORT terminates with a return code 12.

  USER ACTION: Contact BMC Software.

- DPRS012A NO RECORD MATCHING SELECTION CRITERIA:

  REASON: No input statistics records were found that match the criteria specified.

  SYSTEM ACTION: DPREPORT terminates with a return code 16.

  USER ACTION: Determine error in selection criteria or input file.

- DPRS013A INTERNAL ERROR: INVALID IVC FLAG BYTE

  REASON: Internal error.

  SYSTEM ACTION: DPREPORT terminates with a return code 12.

  USER ACTION: Contact BMC Software.

- DPRS014A INVALID VARIABLE DESCRIPTOR BLOCK

  REASON: Internal error in the statistics DDF report definitions.

  SYSTEM ACTION: DPREPORT terminates with a return code 12.

  USER ACTION: Contact BMC Software.

# Chapter 5. DB2 Tables Reporting Facilities

This chapter discusses the facilities to produce reports from the data in DB2 tables.

Reporting flexibility and a historical perspective is achieved by storing the Performance Reporter data within DB2. Performance Reporter provides predefined reports using SQL statements, which can be run through a batch reporting facility (DPRREPT) or through QMF queries. DB2 performance charts and plots are also provided, which are available through QMF only. Other queries and reports based on the performance data tables can also be defined.

The input to this process is any of the performance data tables. Reports, charts, and plots of DB2 performance comprise the output.

## Prepared Reports

The prepared reports included with Performance Reporter show DB2 system workload to help the DB2 performance analyst, capacity planner, or service manager solve specific DB2 problems. These reports are generated from three types of data:

Accounting data

Pertains to the execution of a program, transaction, or plan. The data can be from the detail accounting table (one row per plan execution) or from the summary accounting table (one row per summary interval per unique key). It can show base, DDF, or optional package information.

Statistics data

Shows activity within the system at discrete intervals (DB2 statistics interval). It can show base, DDF, or detail buffer pool information.

Audit data

Shows audit activity for each authorization ID and plan. It can show information on

- Audit summary
- Audited DML Access
- Authorization failures
- Authorization control - GRANTs / REVOKEs
- Audited DDL access
- DML at BIND
- AUTHID change
- Audit utility access
- Audit detail

## Organization

The reports from both accounting and statistics data are organized as follows:

General DB2 System
: Overview and exception reports

User Activity
: SQL, CPU/elapsed time reports, and package reports

DDF Activity
: Local and Remote DB2 Accounting DDF

Locks
: Lock and latch activity

EDM Pool
: EDM pool statistics

Buffer Pools
: Buffer pool overview and detail reports

Logging
: DB2 logging statistics

Audit
: DB2 audit activity

## Naming Convention

A naming format has been assigned to the reports to allow for a faster understanding of the data source and basic orientation of each report.

A report name is constructed as

```
xxzzzzzy
```

where

xx  Data source

| | |
|---|---|
| AC | Detailed accounting data |
| SA | Summarized accounting data |
| ST | Statistics data |
| AU | Audit data |

zzzzz  Report name

For buffer pool or global buffer pool reports, these characters are further refined as:

```
BFpzz (buffer pool), or GBpzz (global buffer pool)
```

where

p  Groups the buffer pools by:

| | |
|---|---|
| A | All buffer pools |
| X | Single buffer pool |

zz  Defines

| | |
|---|---|
| T | Totals |
| A | Averages |
| VP | Virtual pool activity |
| PF | Prefetch activity |
| HP | Hiperpool activity |
| EX | Exceptions (buffer pool or global pool) |
| null | Global pool activity |
| 2 | Extended report |

y  Report orientation

| | |
|---|---|
| P | Plan oriented report (from accounting data) |
| A | AUTHID oriented report (from accounting data) |
| null | Date oriented report (from statistics data) or report oriented by both AUTHID and plan (from audit data) |

**Note:** The QMF query members follow the same naming convention, except they are prefixed with a Q. The corresponding QMF form members are prefixed with an F.

All of the prepared reports provided with this product are listed in the table on page 336. See "Reports from DB2 Tables" on page 333 for detailed examples of these distributed reports and their contents.

## Custom Reports and Charts

Since Performance Reporter uses either QMF or a generalized report generator to provide reporting and charting facilities, the entire report is defined by the SQL SELECT statement used to read the data table(s). Thus, it is very easy to create customized reports for a specific requirement. The SQL select statements used to produce the distributed reports and charts are available, within the QMF or DPRREPT environments, as models in tailoring new reports or charts.

There are some restrictions placed on SQL statements used by DPRREPT to control page formatting. These restrictions are described in “Customizing Reports” on page 83; however, their use is optional.

To construct meaningful SQL SELECT statements to report on performance data tables contents, you should be familiar with the contents of these tables as described in “Performance Data Tables” on page 443. All the rules of programming SQL statements must be followed.

Of course, other DB2 reporting facilities available in-house, such as various fourth generation languages with a DB2 interface, can be used to access the MVDB2 performance data tables.

## Standard Reporting

Standard reporting can be run on a daily and/or weekly basis.

Daily run

The queries processed by the batch reporting program, DPRREPT, from the detail statistics (queries STxxx) and detail accounting (queries ACxxxx) tables are set up to produce a set of daily reports on the data loaded from the previous day (current date minus one). The sample JCL in BBSAMP named DPRRPT includes all distributed reports. Run these reports once, select those ACxxxx and STxxx reports you want to review daily, and create a job for this daily run.

Weekly run

The queries processed by the batch reporting program, DPRREPT, from the summary accounting table (queries SAxxxx) are set up to produce a set of weekly reports on the data from the previous week (current date to current date minus seven). Select the SAxxx reports you want to review weekly and create a job for this weekly run.

## Ad Hoc Reporting

Any of the distributed SQL queries can be modified or used as a model to produce queries to satisfy ad hoc reporting needs. However, the flexibility of QMF, in general, makes it the better vehicle for such reporting. An added advantage with QMF is that the procs provide an easy way to select a specific time period for reporting without modifying the queries themselves.

# Batch Reporting Facility (DPRREPT)

The queries on the detail tables automatically produce a report of yesterday's data. These queries can be modified (see "Customizing Reports" on page 83). The same reports are produced with either the batch reporting facility or with QMF.

## JCL Statements

Producing predefined reports is a single-step procedure, as shown in Figure 8 on page 74.

Following is a description of the job control language (JCL) statements used in Figure 8.

### Batch reporting procedure

REPORT
: Specifies the name of the Performance Reporter batch program as

  PGM=JXREPT

  and the region required to run the program.

STEPLIB DD
: Defines the program library containing the JXREPT module.

JXRPPRT DD
: Defines the diagnostic messages and counts from reporting.

JXRPREP DD
: Defines the requested reports.

SYSUDUMP DD
: Defines the dump data set for problem determination.

JXRPSQL DD
: Defines the BBPARM library containing the report definitions.

JXRPUT1 DD
: Defines the utility work data set for reporting.

JXRPIN DD

Defines the input control statements. The syntax is as follows:

SSID=

Identifies the name of the DB2 system which contains the tables from which the reports are to be generated. The reporting program establishes a CAF connection. Reports may be generated from multiple DB2 systems by re-specifying this value.

PLAN=

Identifies the plan name for reporting. The default is JXREPT.

&DATE=

Defines the date range for reported data. It can include any valid SQL WHERE restrictions in addition to the required DATE operand.

This value replaces the default DATE SQL statement in all following BBPARM members (until another &DATE or &WHERE statement is found). All characters after the equal sign (=) through the last non-blank character on a line are substituted. If the substitution value is longer than one line, put any non-blank character in column 72 and continue the substitution value in column 1 of the next line.

&DATE= supports either direct substitution from the input statement using DB2's CURRENT DATE special register or substituted calculated relative dates.

Relative dates are requested with the use of * for today's date or `*_Nx-M` for a range of dates relative to the current time period as defined by x where.

`x` can be D for Day (default), W for Week, or M for Month

`_N` represents the number of time periods, prior to the current one, to be selected as the *from* value

`-M` is optional and it represents the *up to* value in the time periods as defined by x. The selected data will not include the data relative to the current period of time. If `-M` is not used, the *up to* value is set to the current time period.

**Note:** Use of the calculated relative dates (* forms) is highly recommended and results in index usage with significant performance advantages over using DB2's CURRENT DATE special register.

For example, to generate reports of today's data, use

```
&DATE=*
```

For reports of yesterday's data only, use

```
&DATE=*_1-*
```

or

```
&DATE=*_1D-*
```

For reports of yesterday's data and today's date, use

```
&DATE=*_1
```

For reports of the previous week's data, but not including data from the current week use

```
&DATE=*_1W-*
```

For reports of the previous month's data, but not including any data from the current month, use

```
&DATE=*_1M-*
```

For reports of the previous month’s data for the previous year, use

```
&DATE=*_12M-*_11
```

For reports using the CURRENT DATE special register and DB2 functions with yesterday’s data, use

```
&DATE= = DATE(DAYS(CURRENT DATE)-1)
```

**Note:** The repeated equal sign is not a typographical error in the previous example.

For reports using the CURRENT DATE special register and DB2 functions with the past week’s data, use

```
&DATE= BETWEEN DATE(DAYS(CURRENT DATE)-6) AND CURRENT DATE
```

&CYCLE=
HOURLY|DAILYWEEKLY|MONTHLY|WORKWKLY|WEEKENDS|RESET
Defines an interval for summarizing report data. Report SELECT statements are modified and the GROUP BY clause is used to summarize the data by the interval requested. The summarization interval remains in effect for all following BBPARM members until another &CYCLE statement is processed. &CYCLE=RESET results in cancelling a previous &CYCLE statement. Each time a new interval occurs in the report, the new time is displayed either as a new report line or by replacing an existing DATETIME report column.

Parameters are mutually exclusive except either WORKWKLY or WEEKENDS may be used with the interval selections. WORKWKLY limits report data table records to those created by DB2 on Monday through Friday and WEEKENDS limits report data to Saturday and Sunday.

DATETIME=YES|NO
Requests DATETIME column be added to the GROUP BY clause of the report SELECT statement. This control statement is intended to be used for reports from accounting summarization data and displays the time of the summarized accounting record as a separate report line. DATETIME=YES activates the function and DATETIME=NO cancels it.

&TABLE=
Identifies the table from which reported data is selected. This value replaces the first table name identified in the FROM SQL statement in the BBPARM members.

&TABLE2=
Identifies the table from which reported data is selected. This value replaces the second table name identified in the FROM SQL statement in the BBPARM members.

mbrnames
Keyword indicating the JXRPSQL member name that defines the report to be executed.

You can specify multiple member names on different lines. These members reside in the data set defined by JXRPSQL.

* (asterisk)
When this character is in column one, it specifies a comment.

Figure 8 shows sample job control statements to generate reports.

```
//DPRREPT  JOB
//DPPREPT PROC HIDP='HILVL.RUN.LIB',               <--- UPDATE
//             HIDB2=DSN510,                       <--- VERIFY
//             UNIT=SYSDA                          <--- VERIFY
//*------------------------------------------------------------*
//*            PRODUCE THE PERFORMANCE REPORTER SET            *
//*            OF REPORTS                                      *
//*                                                            *
//*            AC - ACCOUNTING DETAIL REPORTS                  *
//*            SA - ACCOUNTING SUMMARY REPORTS                 *
//*            ST - STATISTICS DETAIL REPORTS                  *
//*            AU - AUDIT REPORTS                              *
//*------------------------------------------------------------*
//REPORT  EXEC PGM=JXREPT,REGION=2048K
//STEPLIB   DD DISP=SHR,DSN=&HIDB2..SDSNLOAD
//          DD DISP=SHR,DSN=&HIDP..BBLINK
//JXRPPRT   DD SYSOUT=*           EXCEPTION/STATUS MESSAGES
//JXRPREP   DD SYSOUT=*           REPORTS
//SYSUDUMP  DD SYSOUT=*
//JXRPSQL   DD DISP=SHR,DSN=&HIDP..BBPARM
//JXRPUT1   DD UNIT=&UNIT,SPACE=(TRK,(40,20)),
//             DCB=BLKSIZE=23440
//JXRPIN    DD DUMMY
//        PEND
//*
//REPORT  EXEC DPPREPT
//JXRPIN    DD *
  SSID=DDDD
  PLAN=JXREPT
* &DATE= = DATE(DAYS(CURRENT DATE)-1)
* &DATE= BETWEEN  DATE(DAYS(CURRENT DATE)-7) AND CURRENT DATE
  &DATE=*_1
```

Figure 8.  Sample DPRREPT Job Control Statements  (Page 1 of 4)

```
  &TABLE=DMRPR.DMRACDTL
*            IF THE DETAIL ACCOUNTING TABLES WERE NOT LOADED IN THE
*            JOB DPRSMF (EXTRACT/SUMMARIZATION JCL STEP), REMOVE OR
*            COMMENT OUT THESE REPORT REQUESTS BEGINNING WITH AC.
*            THE TABLES WILL HAVE NO DATA IN THEM.
   ACOVERA
   ACOVERP
   ACSQLA
   ACSQLP
   ACCPUA
   ACCPUP
   ACBFAAA
   ACBFAAP
   ACBFATA
   ACBFATP
   ACLOCKA
   ACLOCKP
   ACRIDA
   ACRIDP
   ACPLIOA
   ACPLIOP
   ACEXCEP
   ACSPRCA
   ACSPRCP
   ACGLOKA
   ACGLOKP
   ACGBAAA
   ACGBAAP
   ACGBATA
   ACGBATP
   ACDYNCA
   ACDYNCP
  &TABLE=DMRPR.DMRADDTL
   ACOVDFA
   ACOVDFP
   ACCPDFA
   ACCPDFP
   ACDDFLA
   ACDDFLP
   ACDDFRA
   ACDDFRP
  &TABLE=DMRPR.DMRAPDTL
   ACPKGA
   ACPKGP
   ACPKGS
  &TABLE=DMRPR.DMRABDTL
   ACBFXAA
   ACBFXAP
   ACBFXTA
   ACBFXTP
```

Figure 8.  Sample DPRREPT Job Control Statements  (Page 2 of 4)

```
&TABLE=DMRPR.DMRACSUM
 SAOVERA
 SAOVERP
 SASQLA
 SASQLP
 SACPUA
 SACPUP
 SABFAAA
 SABFAAP
 SABFATA
 SABFATP
 SALOCKA
 SALOCKP
 SAPLIOA
 SAPLIOP
 SARIDA
 SARIDP
 SASPRCA
 SASPRCP
 SAGLOKA
 SAGLOKP
 SAGBAAA
 SAGBAAP
 SAGBATA
 SAGBATP
 SADYNCA
 SADYNCP
&TABLE=DMRPR.DMRADSUM
 SAOVDFA
 SAOVDFP
 SACPDFA
 SACPDFP
 SADDFLA
 SADDFLP
 SADDFRA
 SADDFRP
&TABLE=DMRPR.DMRAPSUM
 SAPKGA
 SAPKGP
 SAPKGS
&TABLE=DMRPR.DMRABSUM
 SABFXAA
 SABFXAP
 SABFXTA
 SABFXTP
```

Figure 8.  Sample DPRREPT Job Control Statements  (Page 3 of 4)

```
  &TABLE=DMRPR.DMRSTAT
   STSQL
   STCPU
   STLOCK
   STEDM
   STEDM2
   STLOG
   STRID
   STEXTHD
   STSPROC
   STGLOCK
   STDYNC
   STOVER
   STOVERT
   STBFAEX
   STBFAHP
   STBFAPF
   STBFAVP
   STPLIO
   STEXSYS
   STGBA
   STGBA2
   STGBAEX
  &TABLE=DMRPR.DMRSBFDT
   STBFXEX
   STBFXHP
   STBFXPF
   STBFXVP
   STGBX
   STGBX2
   STGBXEX
  &TABLE=DMRPR.DMRSTDF
   STOVDFT
  &TABLE=DMRPR.DMRAUSUM
   AUSUM
   AUDTL
  &TABLE=DMRPR.DMRAUFAL
   AUFAIL
  &TABLE=DMRPR.DMRAUGRV
   AUDGRV
  &TABLE=DMRPR.DMRAUDDL
   AUDDL
  &TABLE=DMRPR.DMRAUDML
   AUDML
  &TABLE=DMRPR.DMRAUDMB
   AUDMLB
  &TABLE=DMRPR.DMRAUCHG
   AUCHNG
  &TABLE=DMRPR.DMRAUUTL
   AUUTIL
*-----------------------------------------------------------*
//
```

Figure 8. Sample DPRREPT Job Control Statements (Page 4 of 4)

# Batch Reporting Facility Return Codes and Error Messages

The following return codes and error messages are produced by the batch reporting facility (JXREPT).

## Return Codes

The return codes produced by the JXREPT program are

| | |
|---|---|
| **0** | Successful completion |
| **4** | Warning issued |
| **8** | At least one major function was not completed |
| **12** | Catastrophic, no functions completed |

## Error Messages

The following error messages are produced by the JXREPT program. All messages are written to the JXRPPRT data set.

- JXR0101I - counter: value

  REASON: Counts of activity are maintained during the reporting process. At the end of processing, each of the non-zero counters is displayed with this message.

  SYSTEM ACTION: None.

  USER ACTION: None.

- JXR0102C - ERROR IN OPEN FOR ddn

  REASON: The specified DDNAME could not be opened.

  SYSTEM ACTION: Processing ends.

  USER ACTION: Correct and resubmit.

- JXR0103E - CONNECT ERROR TO ddd WITH PLAN ppp - REQUEST SKIPPED

  REASON: An error in the CAF connection was encountered as defined by message JXR0311E. This message shows the target DB2 system and the plan name.

  SYSTEM ACTION: None.

  USER ACTION: None.

- JXR0104I - CONNECTED TO ddd WITH PLAN ppp

  REASON: The CAF connection was successful to target ddd with plan ppp

  SYSTEM ACTION: None.

  USER ACTION: None.

- JXR0105I - LOAD FAILED FOR lll

  REASON: A load request failed for module lll. The possible module names include

  – DSNTIAR
  – DSNHDECP

  SYSTEM ACTION: Processing ends.

  USER ACTION: Correct and resubmit.

- JXR0106C - CANNOT OBTAIN CURRENT DATE - date error msg

  REASON: An error was encountered in requesting current date from the system.

  SYSTEM ACTION: Processing ends.

  USER ACTION: Correct and resubmit.

- JXR0107I - END OF REPORT - nnn PAGES WRITTEN

  REASON: A report request has been successfully processed and nnn pages were written.

  SYSTEM ACTION: None.

  USER ACTION: None.

- JXR0108I - PROCESSING STATISTICS:

  REASON: This heading line proceeds a series of JXR0101I messages.

  SYSTEM ACTION: None.

  USER ACTION: None.

- JXR0121I - INPUT: input image

  REASON: This echoes an input member request, showing the member name, options, and possible comments.

  SYSTEM ACTION: None.

  USER ACTION: None.

- JXR0122E - INVALID MEMBER NAME: mbr

  REASON: The member name mbr is invalid - either too long or does not conform to the rules for member name.

  SYSTEM ACTION: Processing ends for this member.

  USER ACTION: Correct and resubmit.

- JXR0123E - INVALID COLS SPEC ON MEMBER: cols

  REASON: The cols specification on the member name is invalid.

  SYSTEM ACTION: The value is ignored.

  USER ACTION: Correct and resubmit.

- JXR0124I - COMMENTS IN INPUT FOLLOW:

REASON: The lines that follow are comments included in the input stream.

SYSTEM ACTION: None.

USER ACTION: None.

- JXR0125I - SUBSTITUTION CONTROL STATEMENTS FOLLOW:

REASON: The lines that follow are substitution control statements included in the input stream.

SYSTEM ACTION: None.

USER ACTION: None.

- JXR0126E - INVALID SUBSTITUTION SYNTAX

REASON: A substitution control statement as shown above has invalid syntax. Any statement beginning with an ampersand is considered a substitution statement.

SYSTEM ACTION: Processing ends.

USER ACTION: Correct and resubmit.

- JXR0127C - EXPECTED CONTINUATION NOT RECEIVED

REASON: A substitution control statement was coded as continued, but the continuation was not received.

SYSTEM ACTION: A partial value is used.

USER ACTION: Correct and resubmit.

- JXR0128E - VALUE vvv INVALID FOR kkk

REASON: A value for target DB2 system or planname is invalid.

SYSTEM ACTION: Processing ends until the next valid specification is found.

USER ACTION: Correct and resubmit.

- JXR0129I - VALUE FOR nnn IS vvv

REASON: This messages echoes a control statement defining the target system or plan name.

SYSTEM ACTION: None.

USER ACTION: None.

- JXR0141E - MEMBER name NOT FOUND

REASON: The specified ‘name’ was not found in the SQL library.

SYSTEM ACTION: Processing ends for this member.

USER ACTION: Correct and resubmit.

- JXR0142W - TOO MANY SQL LINES BEFORE SEMI-COLON

  REASON: An SQL statement is limited to 32,768 characters; the input member exceeded this limit.

  SYSTEM ACTION: Processing ends for this member.

  USER ACTION: Correct and resubmit.

- JXR0143W - CONTINUATION ON A CONTINUATION

  REASON: A heading line was continued to a second line which also was coded as continued (non-blank in column 72).

  SYSTEM ACTION: The second heading is ignored.

  USER ACTION: Correct and resubmit.

- JXR0144W - TOO MANY HEADING LINES

  REASON: There is a limit of 11 heading lines in a report definition.

  SYSTEM ACTION: The extra lines are ignored.

  USER ACTION: Correct and resubmit.

- JXR0145W - LAST CONTINUATION FOR HEADING MISSING

  REASON: The last heading line in the member was coded as continued, but it was the last line in the member.

  SYSTEM ACTION: Processing continues.

  USER ACTION: Correct and resubmit.

- JXR0146W - INVALID MAXDECWIDTH - value

  REASON: The value of MAXDECWIDTH was invalid - either non-numeric, too many digits, or the number itself was too large.

  SYSTEM ACTION: Processing ends for this member.

  USER ACTION: Correct and resubmit.

- JXR0161W - FIELD xxx HAS UNKNOWN TYPE CODE ccc

  REASON: The SQL field xxx is defined to DB2 with an unknown type code ccc.

  SYSTEM ACTION: Processing continues.

  USER ACTION: Contact BMC Software.

- JXR0162W - PRINT LINE OVERFLOW WITH n FIELDS LEFT

  REASON: The report as defined could not fit on a 133 character report line.

  SYSTEM ACTION: Processing continues with truncated lines.

  USER ACTION: Correct and resubmit.

- JXR0163C - PREPARE FAILED - NO VALUES RETURNED

  REASON: An error was encountered in PREPARing an SQL statement. This is accompanied by message JXR0164E.

  SYSTEM ACTION: Processing ends for this member.

  USER ACTION: Correct and resubmit.

- JXR0164E - SQL FAILURE CODE FOLLOWS:

  REASON: Following is a variable number of lines produced by DSNTIAR detailing the SQL error received.

  SYSTEM ACTION: Processing ends for this member.

  USER ACTION: Correct and resubmit.

- JXR0179W - FIELD xxx HAS UNKNOWN TYPE CODE ccc

  REASON: The SQL field xxx is defined to DB2 with an unknown type code ccc.

  SYSTEM ACTION: Processing ends for this member.

  USER ACTION: Contact BMC Software.

- JXR0181E - FIELD xxx HAS INVALID TYPE CODE ccc

  REASON: The SQL field xxx is defined to DB2 with an unknown type code ccc.

  SYSTEM ACTION: Processing ends for this member.

  USER ACTION: Contact BMC Software.

- JXR0182E - FIELD xxx HAS UNSUPPORTED TYPE CODE ccc

  REASON: The SQL field xxx is defined to DB2 with a type code ccc which is unsupported by the reporting routine.

  SYSTEM ACTION: Processing ends for this member.

  USER ACTION: Contact BMC Software.

- JXR0311E - CAF CALL FAILED - reason.

  REASON: A CAF connection request failed for the stated reason.

  SYSTEM ACTION: Processing ends until the next valid DB2/plan specification is found.

  USER ACTION: Correct and resubmit.

## Customizing Reports

The JXREPT program is designed to run any SQL SELECT statement against the Performance Reporter tables and report the results. See “Prepared Reports” on page 67 for a description of the reports distributed with Performance Reporter. To customize your own reports, see Figure 9 for sample input to DPRREPT.

```
SELECT LOCATION,SUBSYSTEM,XXXXXX,
   FIELD-1,
   FIELD-2,
   FIELD-3,
   FIELD-N,
   MIN(DATETIME),MAX(DATETIME)
       FROM DMRPR.DMRXXXX
       GROUP BY LOCATION,SUBSYSTEM,XXXXXX;
XXXXXXXXXXXXXXXXXXXXXX     ----------------------------TITLE ------X
-----------------------------------------------     XXXXXXXXX
XXXXXXXX                                                 X         X
                             XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX                              X     X
                                                XXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX                               X    X
                                             XXXXXXXXXXXXXXXX
                                                               X   X

----COLUMN HEADER LINE 1-------------------------------------------X
-------------------------------------------------------------
----COLUMN HEADER LINE 2-------------------------------------------X
-------------------------------------------------------------
----COLUMN HEADER LINE 3-------------------------------------------X
-------------------------------------------------------------
```

Figure 9. Sample Input to DPRREPT

## SQL SELECT Statement Recommendations

When customizing, some conventions must be followed to produce consistent results and the complete page format as demonstrated within the distributed reports. The recommendations for SQL SELECT statements are as follows:

- The first two columns returned in the SELECT statement should be LOCATION and SUBSYSTEM, in that order.

  Placing these two columns first in the SELECT statement results in page breaks on LOCATION and SUSBSYSTEM changes. The columns normally produced within the body of the report do not occur when this convention is observed. This information will be reported within the page header lines instead.

  If you do not observe this convention, there will be no special page breaks based on these values. The data, if selected in other than the first two columns, will occur as columns within the body of the report.

- The last two columns in the SELECT statement should be MIN(DATETIME) and MAX(DATETIME), in that order.

  Placing these columns in the SELECT statement results in the range of data encountered within the tables to be reported within the page header. When this convention is followed, these last two columns do not appear in the body of the report. Be careful not to select other timestamp data type columns as the last two columns to avoid incorrect results.

  If you do not observe this convention, blanks appear in place of these date ranges in the page headers.

- When LOCATION and SUBSYSTEM are the first two columns selected, LOCATION and SUBSYSTEM should be the highest sequencing field in an ORDER BY clause in the SELECT statement.

  This procedure allows the reports with page breaks on LOCATION and SUBSYSTEM changes to be properly paginated.

  Failure to follow this convention can result in quite excessive and incorrect pagination for the report.

## Specifying Report Headers

Each sample report has its headers defined in the same member as the SQL SELECT statement, beginning in the line following the SQL completion semicolon (;). Each header line requires two lines in the member. Data is specified in columns 1–71. A non-blank continuation character must be specified in column 72 of the first line of each pair. Headers are 132 characters long; specify 71 characters in the first line and 61 characters in the continuation. All headers are single spaced.

Header fields supplied automatically by the DPRREPT program are indicated with a series of X characters. These fields are

- Header 1 - company name (left), page number (right)
- Header 2 - report name (left), report date (right)
- Header 3 - date from (left), system (right)
- Header 4 - date to (left), subsystem (right)
- Header 5 - blank line

Fields that can be modified are

- Header 1 - title (middle) - blank if not specified
- Headers 6 - 11 - one through six column headers

If no column headers are defined, SQL field names are used where possible and otherwise left blank. When calculating column spacing, observe that one blank is left between each data column.

## Reporting Statistics Data by Time Interval

The default statistics reports show statistics by the DATETIME stamp in each record. To provide a historical view, you may want to modify the distributed reports to show statistics grouped by a longer time interval. To simplify this type of reporting, the statistics tables include columns for several other date and time values, such as DATE, MONTH, DAY, TIME, and HOUR.

**Note:** Data is not *spread* across intervals.

An example of a report by DATE and HOUR is in BBPARM member STOVERH, as shown in Figure 10.

```
BMC SOFTWARE INC                          DB2 STATISTICS OVERVIEW REPORT - AVERAGES                                  PAGE 0002
REPORT: STOVERH                                                                                  REPORT DATE: 2001-03-25 11.14.54
DATE FROM: 2001-03-01 00:09:01                                                                                  LOCATION - SANJOSE
DATE TO:   2001-03-19 23:55:49                                                                                  SUBSYSTEM - DB2P

                CREATE PHASE 2   SYNC            SQL     SQL     SQL   GETPAGES PAGE     READ    WRITE    EDM LOG WRT    LOCK
DATE / HOUR     THREADS COMMITS COMMITS  ABORTS  MANIP. CONTROL DEFINIT   REQS  UPDATES   I/O     I/O     LOADS   CALLS  SUSPEND
--------------- ------- ------- ------- ------- ------- ------- ------- ------- ------- ------ ------- ------- ------- -------
2001-03-01 00         1     0.0     0.0     0.0     0.0     0.0     0.0     6.0     0.0    1.0     0.0     0.0     0.0     0.0
2001-03-01 01         0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0    0.0     0.0     0.0     0.0     0.0
2001-03-01 02         8     0.6     0.5     0.1     1.2     2.2     0.0   367.7   176.7   15.7     1.3     1.3     4.0     0.0
2001-03-01 03         3     0.3     1.0     0.0     0.6     0.0     0.0    64.6    30.6    1.6     4.0     0.0     2.0     0.0
2001-03-01 04         1     0.0     1.0     0.0   253.0     0.0     0.0   520.0     0.0   15.0     0.0     2.0     0.0     0.0
2001-03-01 05         1     2.0     0.0     0.0     1.0     0.0     0.0    10.0     0.0    8.0     0.0     0.0     0.0     0.0
2001-03-01 06        14     1.0     0.0     0.0     1.0     0.8     0.0    17.3     0.0    4.8     0.0     0.2     0.0     0.0
2001-03-01 07        10     0.6     0.7     0.0     1.4     1.5     0.0   304.6   168.1    4.7     2.4     0.0     3.4     0.0
```

Figure 10. Statistics Overview Report by DATE and HOUR

The following changes were made to the STOVER report to create this example, and similar changes can be made to any other statistics report:

- In the SELECT statement, add the desired columns (A.DATE and A.HOUR) after A.SUBSYSTEM and before the selected data columns, replacing the substring of DATETIME
- Change the Group By clause to

  ```
  GROUP BY A.LOCATION,A.SUBSYSTEM,A.DATE,A.HOUR
  ```
- Optionally, change the column header DATETIME to DATE / HOUR
- Change the WHERE clause with parameters at runtime to select the time period reported on

## Changing the Order of Audit Reports

The sample audit reports are ordered by primary authorization ID and plan name. However, you can easily change this order to any of the following identifiers:

| | |
|---|---|
| CONNECT | Connection ID |
| CORRNAME | Correlation name |
| CORRNMBR | Correlation number |
| ORIGAUTH | Original authorization ID |
| PLANNAME | Plan name |
| PRIMAUTH | Primary authorization ID |
| REQLOC | Requesting location |

## Using QMF to Produce Reports and Charts

QMF can be used to run distributed reports or tailor custom performance reports and charts. A generalized QMF procedure, DZPRQRPT, runs predefined Performance Reporter queries and formats the reports using the distributed Performance Reporter forms. The predefined reports produced with QMF are the same as those produced with the batch reporting facility, DPRREPT. However, with QMF, you can select a range of dates for these reports.

You must enter the QMF program to use this procedure. Detailed instructions to run and use QMF can be found in the *IBM Query Management Facility Learner’s Guide*.

## Running Predefined Queries

To run the predefined Performance Reporter queries from QMF, type the command:

```
RUN DMRPR.DZPRQRPT (&REPORT=report name
```

where `report name` is the name of the report you want to produce. For example, to run the DB2 Accounting Overview Report, type

```
RUN DMRPR.DZPRQRPT (&REPORT=ACOVERA
```

## Defining Table Qualifier and Date Ranges

The report table date range prompts FROM and TO might appear when running these queries under QMF. Type the date range you want using the standard TSI format:

```
'YYYY-MM-DD'
```

Or, you can set global report variables by typing one of these procs:

```
RUN DMRPR.DZPRQDAT
```

```
RUN DMRPR.DZPRQDEF
```

Once you have set these global variables, they will be used by all Performance Reporter report queries for the duration of the QMF session.

## Removing Objects from QMF Tables

To remove all Performance Reporter objects from your QMF tables:

1. Type QMF and import the proc, DZPRQDEL, from BBSAMP:

   ```
   IMPORT PROC FROM ‘hilevel.BBSAMP (DZPRQDEL)’
   ```

2. Press PF2 to run the proc or type the command:

   ```
   RUN PROC
   ```

## Producing QMF Batch Reports

To produce QMF batch reports, use the sample jobs in BBSAMP member DZPRQBAT.

# Part 2. Reports from SMF

This part describes each of the performance reports produced from SMF data as follows:

- A sample report
- Definitions for each field in the report

The JCL to produce these reports is described in Chapter 4, “SMF Reporting Facilities (DPREPORT)” on page 57.

# Chapter 6. Report Format

The reports from SMF provided by Performance Reporter contain a common format for the report page heading, as described in this chapter.

The general page format is shown in Figure 11.

```
LOCATION: DB0G         GROUP : N/A              MAINVIEW FOR DB2                 INTERVAL START: 2001-01-27-15.27.11.6109
SSID    : DB0G         MEMBER: N/A            PERFORMANCE REPORTER               INTERVAL END  : 2001-03-02-08.15.40.7926
VERSION : 41           SCOPE : N/A           ACCOUNTING SHORT REPORT             INTERVAL      : 00001     PAGE 0001/0001
PRAUTH  : BOLBPL1      PLANAM: RXDB2                                             TOTAL RECORDS : 00660
```

Figure 11. Sample Performance Reporter SMF Report Page Layout

The headings shown on each of the reports are described below.

**Note:** Data fields not applicable to a version are shown as N/A.

LOCATION

DB2 local location name (QWHSLOCN). A page break occurs when this field changes.

Positioned on left side of line 1 of report page.

SSID

DB2 subsystem producing data contained in the following report pages (QWHSSSID). A page break occurs when this field changes.

Positioned on left side of line 2 of report page.

VERSION

Version and release of the DB2 subsystem shown in SSID (QWHSRN).

Positioned on left side of line 3 of report page.

GROUP

DB2 data sharing group name (QWHADSGN).

Positioned on left side of line 1 of report page, following LOCATION.

MEMBER

DB2 member name within the DB2 data sharing group (QWHAMEMN).

Positioned on left side of line 2 of report page, following SSID.

SCOPE

Scope of the report as specified by the RSCOPE keyword. It can be

GROUP

Shows data by combination of location and group. Whenever either of these values changes, a page break occurs.

MEMBER

Shows data by a combination of location, group, subsystem, and members. Whenever one of these values changes, a page break occurs.

N/A

Data is for a DB2 subsystem that is not part of a data sharing group.

Positioned on left side of line 3 of report page, following VERSION.

KEY1 (accounting reports only)

First accounting summary keyword and value, if specified. KEY1 may be CONNID, CORRID, ORAUTH, PLANAM, PRAUTH, REQLOC, CONTYP, or PAKAGE, based on the specification of key1 in the RORDER statement. The keyword is followed by the current summarization value. A page break occurs when this field changes.

Positioned on left side of line 4 of report page.

KEY2 (accounting reports only)

Second accounting summary keyword and value, if specified. KEY2 may be CONNID, CORRID, ORAUTH, PLANAM, PRAUTH, REQLOC, CONTYP, or PAKAGE, based on the specification of key2 in the RORDER statement. The keyword is followed by the current summarization value. A page break occurs when this field changes.

Positioned on left side of line 4 of report page, following KEY1.

product name

MVDB2 denotes the product name.

Positioned in center of line 1 of report page.

component name

Performance Reporter denotes the component name.

Positioned in center of line 2 of report page.

title

Report title that describes the nature of the data contained within this report.

Positioned in center of line 3 of report page.

INTERVAL START

Start datetime range of data included in this reporting interval.

Positioned on right side of line 1 of report page.

INTERVAL END

Ending datetime range of data included in this reporting interval.

Positioned on right side of line 2 of report page.

INTERVAL

Number of this reporting interval. A page break occurs when this field changes.

Positioned on right side of line 3 of report page.

PAGE

Page number within the total number of pages for this reporting interval or summarization.

Positioned on right side of line 3 of report page.

TOTAL RECORDS

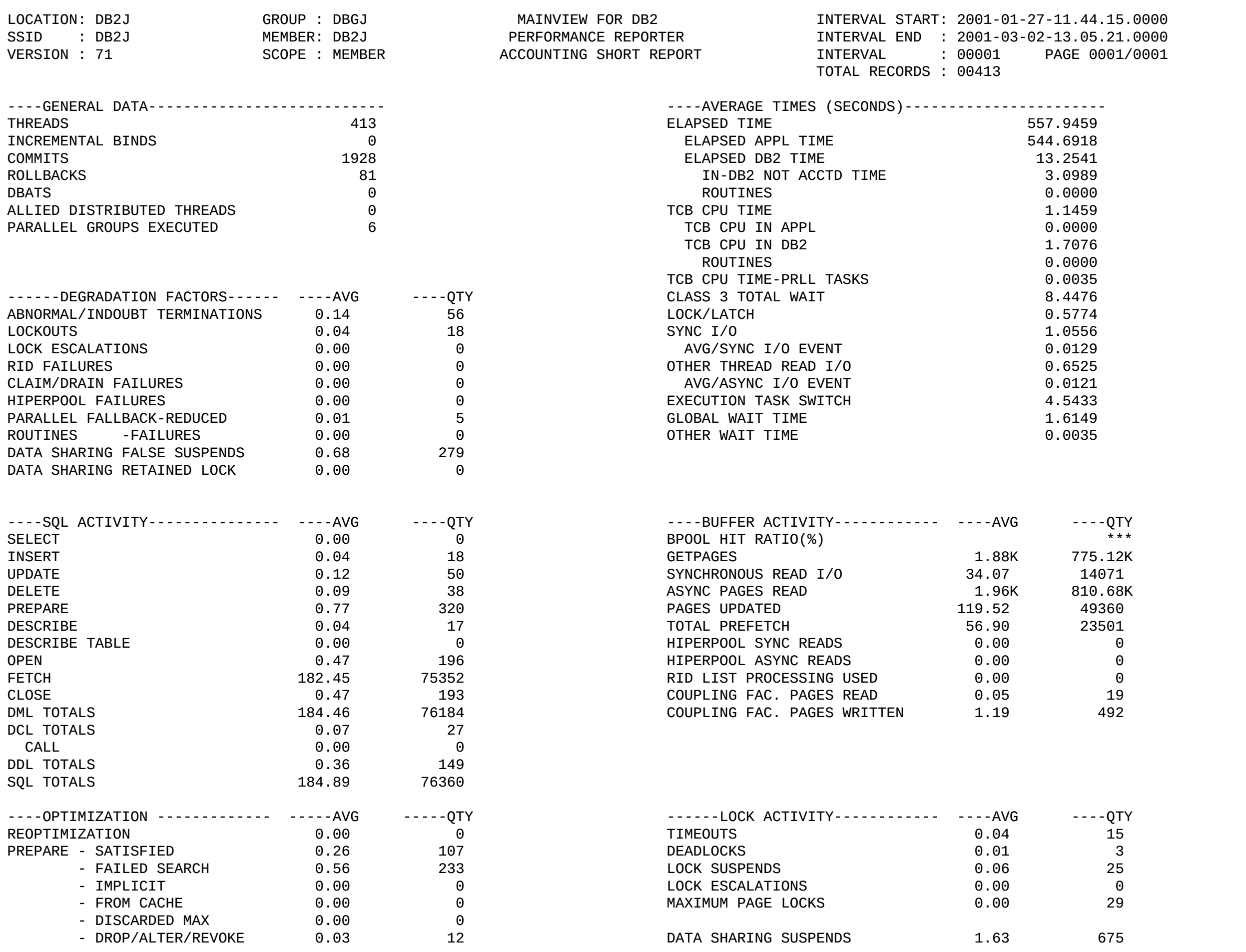
Total number of records included in this reporting interval or summarization.

Positioned on right side of line 4 of report page.

# Chapter 7. Accounting Short Report

This chapter describes the accounting short report, as shown in Figure 12–Figure 14. It has one page per summary key, followed by pages for package accounting and DDF information, if available.

**Note:** Values displayed in the AVG columns are calculated by dividing the total events or times by the number of threads.



```
LOCATION: DB2J               GROUP : DBGJ                 MAINVIEW FOR DB2                  INTERVAL START: 2001-01-27-11.44.15.0000
SSID    : DB2J               MEMBER: DB2J                PERFORMANCE REPORTER               INTERVAL END  : 2001-03-02-13.05.21.0000
VERSION : 71                 SCOPE : MEMBER             ACCOUNTING SHORT REPORT             INTERVAL      : 00001     PAGE 0001/0001
                                                                                            TOTAL RECORDS : 00413

----GENERAL DATA---------------------------                                ----AVERAGE TIMES (SECONDS)-----------------------
THREADS                                413                                 ELAPSED TIME                              557.9459
INCREMENTAL BINDS                        0                                   ELAPSED APPL TIME                       544.6918
COMMITS                               1928                                   ELAPSED DB2 TIME                         13.2541
ROLLBACKS                               81                                     IN-DB2 NOT ACCTD TIME                   3.0989
DBATS                                    0                                     ROUTINES                                0.0000
ALLIED DISTRIBUTED THREADS               0                                 TCB CPU TIME                                1.1459
PARALLEL GROUPS EXECUTED                 6                                   TCB CPU IN APPL                           0.0000
                                                                             TCB CPU IN DB2                            1.7076
                                                                               ROUTINES                                0.0000
                                                                           TCB CPU TIME-PRLL TASKS                     0.0035
------DEGRADATION FACTORS------  ----AVG      ----QTY                      CLASS 3 TOTAL WAIT                          8.4476
ABNORMAL/INDOUBT TERMINATIONS      0.14           56                       LOCK/LATCH                                  0.5774
LOCKOUTS                           0.04           18                       SYNC I/O                                    1.0556
LOCK ESCALATIONS                   0.00            0                         AVG/SYNC I/O EVENT                        0.0129
RID FAILURES                       0.00            0                       OTHER THREAD READ I/O                       0.6525
CLAIM/DRAIN FAILURES               0.00            0                         AVG/ASYNC I/O EVENT                       0.0121
HIPERPOOL FAILURES                 0.00            0                       EXECUTION TASK SWITCH                       4.5433
PARALLEL FALLBACK-REDUCED          0.01            5                       GLOBAL WAIT TIME                            1.6149
ROUTINES     -FAILURES             0.00            0                       OTHER WAIT TIME                             0.0035
DATA SHARING FALSE SUSPENDS        0.68          279
DATA SHARING RETAINED LOCK         0.00            0


----SQL ACTIVITY---------------  ----AVG      ----QTY                      ----BUFFER ACTIVITY------------  ----AVG      ----QTY
SELECT                             0.00            0                       BPOOL HIT RATIO(%)                                ***
INSERT                             0.04           18                       GETPAGES                           1.88K      775.12K
UPDATE                             0.12           50                       SYNCHRONOUS READ I/O              34.07        14071
DELETE                             0.09           38                       ASYNC PAGES READ                   1.96K      810.68K
PREPARE                            0.77          320                       PAGES UPDATED                    119.52        49360
DESCRIBE                           0.04           17                       TOTAL PREFETCH                    56.90        23501
DESCRIBE TABLE                     0.00            0                       HIPERPOOL SYNC READS               0.00            0
OPEN                               0.47          196                       HIPERPOOL ASYNC READS              0.00            0
FETCH                            182.45        75352                       RID LIST PROCESSING USED           0.00            0
CLOSE                              0.47          193                       COUPLING FAC. PAGES READ           0.05           19
DML TOTALS                       184.46        76184                       COUPLING FAC. PAGES WRITTEN        1.19          492
DCL TOTALS                         0.07           27
  CALL                             0.00            0
DDL TOTALS                         0.36          149
SQL TOTALS                       184.89        76360

----OPTIMIZATION -------------  -----AVG     -----QTY                      ------LOCK ACTIVITY------------  ----AVG      ----QTY
REOPTIMIZATION                     0.00            0                       TIMEOUTS                           0.04           15
PREPARE - SATISFIED                0.26          107                       DEADLOCKS                          0.01            3
        - FAILED SEARCH            0.56          233                       LOCK SUSPENDS                      0.06           25
        - IMPLICIT                 0.00            0                       LOCK ESCALATIONS                   0.00            0
        - FROM CACHE               0.00            0                       MAXIMUM PAGE LOCKS                 0.00           29
        - DISCARDED MAX            0.00            0
        - DROP/ALTER/REVOKE        0.03           12                       DATA SHARING SUSPENDS              1.63          675
```

Figure 12. Accounting Short Report—Page 1

For a complete description of each section, see

- “GENERAL DATA” on page 99
- “DEGRADATION FACTORS” on page 100
- “SQL ACTIVITY” on page 107
- “OPTIMIZATION” on page 109
- “AVERAGE TIMES (SECONDS)” on page 110
- “BUFFER ACTIVITY” on page 115
- “LOCK ACTIVITY” on page 119

```
 LOCATION: DB2J                GROUP : DBGJ               MAINVIEW FOR DB2               INTERVAL START: 2001-01-27-11.56.59.2758
 SSID    : DB2J                MEMBER: DB2J              PERFORMANCE REPORTER            INTERVAL END  : 2001-03-02-13.05.21.6690
 VERSION : 71                  SCOPE : MEMBER           ACCOUNTING SHORT REPORT          INTERVAL      : 00001     PAGE 0001/0001
                                                                                         TOTAL RECORDS : 00142



 PROGRAM NAME   TYPE       OCCURS     SQLSTMT  CLASS7 ELAPSE TIME     CLASS7 TCB TIME    CLASS8 SUSP TIME      SYSEVENT
 DPSPURGD       DBRM            2       20.00             207.0102              0.5441            206.1772           351
 DPSQLDAD       DBRM            9      263.56              26.1583              0.5320              0.7160           191
 DSN@EP2L       PKG            15        3.40              85.7050              1.4418             78.9571         14658
 DSNESM68       PKG            52       24.56              25.9166              0.1400              3.9757         44853
 DSNTIAD        DBRM           19       22.84               9.1506              0.5000              7.9665          1399
 JXRDSQL        DBRM            1    72256.00             298.9960             20.0569            274.9269         20446
 RXSEL1M        PKG            49        9.12              17.6105              0.0181             17.1900           512
```

Figure 13. Accounting Short Report—Packages

For a complete description of each section, see

- “Accounting Short Report—Packages” on page 122

```
LOCATION: DB0G            GROUP : N/A              MAINVIEW FOR DB2                  INTERVAL START: 2001-01-27-14.09.55.7658
SSID    : DB0G            MEMBER: N/A             PERFORMANCE REPORTER               INTERVAL END  : 2001-03-02-16.31.26.8601
VERSION : 71              SCOPE : N/A            ACCOUNTING SHORT REPORT             INTERVAL      : 00001     PAGE 0001/0001
                                                                                     TOTAL RECORDS : 00067

REMOTE LOC       TR SENT     TR RECV     SQL SENT     SQL RECV     ROW SENT     ROW RECV     CONV-ALLOC    CONV-TERM  ELAPSE REMOTE WAIT
DB0G             0.85        0.00          30.31        0.00         0.00        27.08         0.85         0.00               22.2550
DB1F             1.00        0.00           2.25        0.00         0.00         2.25         1.00         0.00                1.3814
DB2F             0.96        0.02           1.76        0.15         0.07        20.30         0.87         0.00                1.0920
```

Figure 14. Accounting Short Report—DDF

For a complete description of each section, see

- “Accounting Short Report—DDF” on page 123

## Accounting Short Report—Page 1

The fields in this section are organized in the following categories:

### GENERAL DATA

THREADS
Number of thread executions.

INCREMENTAL BINDS
Number of incremental BINDs performed (QXINCR).

This counter is incremented every time a plan is run that was bound with the VALIDATE(RUN) option.

**Tuning Tip:** It is generally undesirable to bind a plan with the VALIDATE(RUN) option since all SQL statements must be rechecked for syntax, authority, and access path every time the plan is executed. VALIDATE(RUN) is required if the program is going to CREATE TABLES (for example, work tables) during the execution, or if testing is required on a piece of code for which the objects do not yet exist. Otherwise, VALIDATE(RUN) should be avoided, as the cost is nearly that of dynamic SQL.

COMMITS
Number of commits processed (QWACCOMM).

This is a count of the number of commits taken for this thread. It includes both phase 2 and single-phase commits.

**Tuning Tip:** This is a useful number to observe while tracking a long job with a known commit frequency. The commits counted are successful.

ROLLBACKS
Number of rollbacks processed (QWACABRT).

This number represents the total number of rollbacks including application abends, application explicit ROLLBACK, application deadlock, application CANCELed by operator or abended due to DB2 resource shortage.

DBATS
Number of Database Access Threads executed.

ALLIED DISTRIBUTED THREADS
Number of allied distributed threads executed.

PARALLEL GROUPS EXECUTED
Total number of parallel groups executed (QXTOTGRP).

## DEGRADATION FACTORS

ABNORMAL/INDOUBT TERMINATIONS

Number of times a thread terminated for abnormal reasons (QWACRINV).

This is a count of the number of thread terminations that are considered to be error conditions.

LOCKOUTS

Total number of lockouts (QTXADEA + QTXATIM).

This is the total number of lockouts due to each of these types of lock contention:

- Deadlocks (QTXADEA)

  This counter is incremented every time DB2 encounters a deadlock situation for which the IRLM must cancel a task involved in a *deadly embrace*.

  **Tuning Tip:** Deadlocks are caused by two threads requesting access to two resources which can never be resolved. DB2 chooses its victim by selecting the thread which has done the least number of updates. It records the deadlock in the MSTR address job log. In well-tuned systems, this number should be low. The most frequent cause of deadlock situations is ascending key indexes. The Type 2 indexes available in DB2 4.1 can resolve many of these problems because no index locks are taken.

- Lock timeouts (QTXATIM)

  This number is incremented every time a DB2 thread waits to get a resource longer than the timeout interval. It is specified with the DSNZPARM IRLMRWT on installation panel DSNTIPI. By default it is 60 seconds. Utilities can be allowed to wait several multiples of IRLMRWT.

  **Tuning Tip:** Lock timeouts are caused by the application failing to commit in time for the other thread to gain access to data on pages. Often the problem can be resolved by reducing the time between commits and putting statements that can be updated near their COMMIT logic. Every time a timeout occurs, DB2 writes the holder and suspender to the MSTR job log. Normally, this number should be as close to zero as possible.

LOCK ESCALATIONS

Total number of lock escalations to shared mode or exclusive mode (QTXALES + QTXALEX):

- Lock escalations to shared mode (QTXALES)

  This counter is incremented every time the number of locks against a single table space exceeds the number set in DSNZPARM NUMLKTS on installation panel DSNTIPJ or the number set in the LOCKMAX clause of the CREATE TABLESPACE statement (DB2 4.1 and later).

  **Tuning Tip:** This is not a normal situation unless using repeatable read. If it occurs often, consider changing the LOCKSIZE or LOCKMAX to a higher value, or consider binding the plan with cursor stability or uncommitted read (UR).

- Lock escalations to exclusive mode (QTXALEX)

  This counter is incremented every time the number of updateable locks against a single table space exceeds the DSNZPARM NUMLKTS on installation panel DSNTIPJ or in the LOCKMAX clause of the CREATE TABLESPACE statement. It occurs when the LOCKSIZE parameter is specified as ANY and DB2 has escalated the lock owner to an exclusive lock of the entire table.

  **Tuning Tip:** This is an extremely undesirable situation, usually caused by leaving the LOCKSIZE(ANY) default. To resolve this situation, consider changing the parameter to LOCKSIZE (PAGE) or, in special situations, LOCKSIZE(ROW). This will cause the offending application to take the -904 unavailable resource error rather than cause general unavailability to the rest of the users. This is almost always caused by application failure to commit in a timely fashion and can be resolved by application code changes as well as by DBA action.

RID FAILURES

Number of RID pool failures due to exceeding internal limits or to no storage (QXMRMIAP + QXNSMIAP):

- RID pool failures due to exceeding internal limits (QXMRMIAP)

  This counter is incremented when DB2 detects that a single index would exceed 25% availability in the RID pool for a multiple index access or list prefetch request. The field can be incremented at any stage (retrieval, sorting, ANDing/ORing of RID lists for non-matching index scan processing) or incremented when each index of a multiple index access scan finds no more storage is available.

  **Tuning Tip:** DB2 attempts to protect the RID pool from a large result set that would exceed 25% of the pool; in other words, it will cause a RID pool failure if the number of RIDS from any single index or any index involved in a multiple index access attempt exceeds 25%. If this happens often, attempt to find the offending application and change the SQL or increase the size of the RID pool.

- RID pool failures due to no storage (QXNSMIAP)

  This counter is incremented when DB2 detects that no storage is available in the RID pool for a multiple index access or list prefetch request. The field can be incremented at any stage (retrieval, sorting, ANDing/ORing of RID lists for non-matching index scan processing) or incremented when each index of a multiple index access scan finds no more storage is available.

  **Tuning Tip:** This is an extremely undesirable situation as the portion of the query which would use the index falls back to table scan processing. If this occurs often, the size of the RID pool specified in SPRMRDP of the DSNZPARMs (on installation panel DSNTIPC) should be increased.

CLAIM/DRAIN FAILURES

Number of unsuccessful claim or drain requests (QTXACLUN + QTXADRUN):

- Unsuccessful claim requests (QTXACLUN)

  This number is incremented every time a user issues a request for a claim to an SQL resource but cannot acquire one, usually because a utility or command DRAIN is on the object being sought.

  **Tuning Tip:** This number is of some significance in determining contention between SQL and other types of utilities or commands.

- Unsuccessful drain requests (QTXADRUN)

  This counter is incremented when a potential drainer (utility or command) cannot obtain use of a page set because the claim count has not dropped to zero within the utility timeout value set in IRLMWRT of DSNZPARMs.

  **Tuning Tip:** This number is of significance in determining the number of unsuccessful utility and command processes due to user activity on the resource.

HIPERPOOL FAILURES

Number of synchronous hiperpool reads that fail due to MVS page stealing or number of pages not written to hiperpool because of expanded storage shortage (QBACHRF + QBACHWF):

- Synchronous hiperpool reads that fail due to MVS page stealing (QBACHRF)

  This number is incremented when DB2 finds a page in the hiperpool but MVS has stolen the buffer.

  **Tuning Tip:** This occurs only if the CASTOUT parameter for that buffer pool is set to YES. DB2 must then reread the page from DASD. If the number is consistently high, it indicates the hiperpool has serious competition for MVS resources.

  It would be appropriate to reduce the size of the hiperpool unless the application is so critical that it is deemed necessary to change CASTOUT to NO. CASTOUT NO essentially takes pages from expanded storage and makes them unavailable to MVS. This is generally not desirable as it may seriously degrade MVS performance for other workloads.

- Pages not written to hiperpool because of expanded storage shortage (QBACHWF)

  This number is incremented when a shortage of hiperpool space exists and DB2 cannot move the pages from central to expanded storage. This occurs when serious MVS demands are stealing the hiperpool pages. When this occurs, pages are aged out from the central virtual buffer pool as they would have without a hiperpool.

  **Tuning Tip:** Consider reducing the size of the hiperpool so that all hiperpools can co-exist with MVS requirements.

PARALLEL FALLBACK-REDUCED

Number of times parallel processing occurred less than planned or a parallel group fell back to sequential processing due to buffer shortage, updateable cursor, lack of sort assist, or unavailable enclave services (QXREDGRP + QXDEGBUF + QXDEGCUR + QXDEGESA + QXDEGENC):

- Parallel group degraded due to buffer shortage (QXREDGRP)

  This counter is incremented when the buffer pool does not have enough buffers to support as many degrees of parallel processing as had been planned. DB2 checks buffer allocations at both BIND and execution time. It assumes there will be buffers set aside for parallel processing. If at execution time a similar number of buffers do not exist, DB2 will degrade the parallel processes to a lesser degree or no parallelism.

  **Tuning Tip:** Three parameters can be altered (ALTER BUFFERPOOL command) to resolve this situation. The overall size of the buffer pool is controlled by the VPSIZE (virtual pool size) parameter. The amount of sequential buffers is set by the VPSEQT (virtual pool sequential threshold) parameter. In the amount of buffers reserved for sequential processing, a reserve of buffers available for parallel processing must be maintained by the VPPSEQT (virtual pool parallel sequential threshold) parameter. If a significant number of parallel processes are degraded due to buffer shortage or contention, consider using a different buffer pool or altering the buffer pool used so that sufficient parallel sequential buffers are present.

- Parallel group fallback to sequential due to buffer shortage (QXDEGBUF)

  This counter is incremented when the buffer pool does not have enough buffer storage in the virtual pool to support parallel processing. DB2 checks buffer allocations at both BIND and execution time. It assumes there will be buffers set aside for parallel processing. If at execution time a similar number of buffers do not exist, DB2 will degrade the parallel processes to a lesser degree or to no parallelism.

  **Tuning Tip:** Three parameters can be altered (ALTER BUFFERPOOL command) to resolve this situation. The overall size of the buffer pool is controlled by the VPSIZE (virtual pool size) parameter. The amount of sequential buffers is set by the VPSEQT (virtual pool sequential threshold) parameter. In the amount of buffers reserved for sequential processing, a reserve of buffers available for parallel processing must be maintained by the VPPSEQT (virtual pool parallel sequential threshold) parameter. If a significant number of parallel processes are degraded due to buffer shortage or contention, consider using a different buffer pool or altering the buffer pool used so that sufficient parallel sequential buffers are present.

- Parallel group fallback to sequential due to updateable cursor (QXDEGCUR)

  This counter is incremented when DB2 detects a cursor that is not clearly read-only and falls back from parallel processing to sequential.

  **Tuning Tip:** To resolve this problem, the application program should have a cursor which is unambiguously read-only, with updates through another cursor or statement.

- Parallel group fallback to sequential due to lack of sort assist (QXDEGESA)

  This counter is incremented when DB2 detects that the hardware sort assist facility is not present to logically partition the DB2 temporary DSNDB07 workfiles. The parallel sort operation falls back to sequential.

  **Tuning Tip:** This situation will occur until the necessary hardware assist is purchased.

- Parallel group fallback to sequential due to unavailable enclave services (QXDEGENC)

  This counter is incremented when DB2 detects that MVS 5.2 enclave support is unavailable to support parallel CP processing. The parallel group falls back to sequential.

  **Tuning Tip:** MVS Enclave Support in MVS 5.2 sets objectives for parallel services to perform within service goals set by management in the MVS Workload Manager. The DB2 parallel tasks run as enclave SRBs. The solution is to migrate this system to MVS 5.2 as soon as practical to do so.

ROUTINES -FAILURES

Number of stored procedure, user-defined function, or trigger routines that failed (QXCALLAB + QXCALLTO + QXCALLRJ + QXCAUDAB + QXCAUDTO + QXCAUDRJ + QXTRGERR):

- Stored procedure abends (QXCALLAB)

  This counter is incremented when a stored procedure abends.

  **Tuning Tip:** Large numbers here indicate that one or more stored procedures have fatal errors.

- SQL CALL statements timed out (QXCALLTO)

  This counter is incremented when the Stored Procedures Address Space does not have a TCB available within the time limit specified in the DSNZPARM TIMEOUT parameter to schedule the execution of an SQL CALL statement.

  **Tuning Tip:** Appropriate fixes include:

  – Reduce the amount of CPU service units in other stored procedures to get better throughput.

  – Stop stored procedures no longer needed.

  – Increase the number of TCBs available in the Stored Procedures Address Space (by increasing the NUMTCB parameter in the Stored Procedures Address Space JCL).

- SQL CALL statements rejected because procedure was stopped (QXCALLRJ)

  This counter is incremented when an SQL application CALLs a stored procedure which has been stopped by the operator or system administrator.

  **Tuning Tip:** If this happens frequently, be sure the procedure is started in normal mode.

- User-defined function abends (QXCAUDAB) (DB2 6.1 and later)

- User-defined functions timed out (QXCAUDTO) (DB2 6.1 and later)

- User-defined functions rejected (QXCAUDRJ) (DB2 6.1 and later)

- SQL errors that occurred during the execution of a triggered action (QXTRGERR) (DB2 6.1 and later)

DATA SHARING FALSE SUSPENDS

Number of false global suspensions (QTGAFLSE).

This counter is incremented every time MVS Cross-System Services detects contention on the resource hash class but not on the resource itself.

**Tuning Tip:** MVS Cross-System Services uses a hash table for efficient determination of whether a resource is locked. This hash table points to a number of synonyms. When contention exists on the hash class but not on the actual object, this is false contention. The more resources declared to be of intersystem interest, the more chances there are of this condition occurring. The CF lock structure may be too small.

DATA SHARING RETAINED LOCK

Number of global requests denied because of incompatible retained lock (QTGADRTA).

This number is incremented every time Global Lock Services denies a lock request because an incompatible lock type has been retained for the requested resources.

This number reflects the instances in which intersystem access to a page or row cannot be obtained because another thread on another system had access to the resource requested but the other system failed. The CF retains locks until the other system has come up to complete the unit of work.

## SQL ACTIVITY

SELECT
Number of embedded SELECT statements (QXSELECT).

This is a count of the number of single embedded SELECT statements processed for the application.

INSERT
Number of INSERT statements (QXINSRT).

This counter is incremented when an INSERT statement is processed for the application.

UPDATE
Number of UPDATE statements (QXUPDTE).

This counter is incremented when an UPDATE statement is processed for the application.

DELETE
Number of DELETE statements (QXDELET).

This counter is incremented when a DELETE statement is processed for the application.

PREPARE
Number of PREPARE statements (QXPREP).

This counter is incremented when the SQL PREPARE statement (which mini-binds a dynamic SQL statement) is processed for the application.

The count between a server and requester may not be equal.

DESCRIBE
Number of DESCRIBE statements (QXDESC).

This counter is incremented when a DESCRIBE statement (for dynamic or distributed processing) is processed for the application.

The count between a server and requester may not be equal.

DESCRIBE TABLE
Number of DESCRIBE TABLE statements (QXDSCRTB).

This counter is incremented every time a DESCRIBE TABLE statement is executed.

OPEN
Number of OPEN CURSOR statements (QXOPEN).

This counter is incremented when the OPEN CURSOR statement is processed for the application.

FETCH
Number of FETCH statements (QXFETCH).

This counter is incremented every time a FETCH statement is executed from a cursor.

CLOSE

Number of CLOSE CURSOR statements (QXCLOSE).

The counter is incremented when an SQL CLOSE CURSOR statement is processed for the application.

The count between a server and requester may not be equal.

DML TOTALS

Total number of SQL DML statements processed for the application.

DCL TOTALS

Total number of SQL DCL statements processed for the application, including stored procedure call statements.

CALL

Total number of SQL stored procedure CALL statements processed for the application.

DDL TOTALS

Total number of SQL DDL statements processed for the application.

SQL TOTALS

Total number of SQL statements processed for the application.

## OPTIMIZATION

These fields apply only to DB2 5.1 and later.

REOPTIMIZATION
Number of times reoptimization occurred for a query (QXSTREOP).

PREPARE - SATISFIED
Number of times DB2 satisfied a PREPARE request by making a copy of a statement in the prepared statement cache (QXSTFND).

PREPARE - FAILED SEARCH
Number of times DB2 searched the prepared statement cache but could not find a suitable prepared statement (QXSTNFND).

PREPARE - IMPLICIT
Number of times DB2 did an implicit PREPARE for a statement bound with KEEPDYNAMIC(YES) because the prepared statement cache did not contain a valid copy of the prepared statement (QXSTIPRP).

PREPARE - FROM CACHE
Number of times DB2 did not prepare a statement bound with KEEPDYNAMIC(YES) because the prepared statement cache contained a valid copy of the prepared statement (QXSTNPRP).

PREPARE - DISCARDED MAX
Number of times DB2 discarded a prepared statement from the prepared statement cache because the number of prepared statements in the cache exceeded the value of subsystem parameter MAXKEEPD (QXSTDEXP).

PREPARE - DROP/ALTER/REVOKE
Number of times DB2 discarded a prepared statement from the prepared statement cache because a program executed a DROP, ALTER, or REVOKE statement against a dependent object (QXSTDINV).

## AVERAGE TIMES (SECONDS)

ELAPSED TIME

Elapsed time from the first connect to DB2 to thread termination (QWACESC - QWACBSC).

This value is accurate with the exception of threads that do not terminate, such as a CICS conversational transaction or an IMS WFI BMP. These include the time the thread was inactive waiting for work.

ELAPSED APPL TIME

Elapsed time NOT in DB2 (QWACESC - QWACBSC - QWACASC).

This value is collected when accounting class 2 is activated.

Application time is elapsed time spent in the application while NOT performing DB2 work.

ELAPSED DB2 TIME

Elapsed time in DB2 (QWACASC).

This value is collected when accounting class 2 is activated.

In-DB2 time is elapsed time while performing DB2 work. The DB2 time is a percentage of the total elapsed time.

IN-DB2 NOT ACCTD TIME

Elapsed time in DB2 not accounted for in either CPU time or total wait time (QWACASC - QWACAJST - CLASS 3 TOTAL WAIT).

This value is collected when accounting classes 2 and 3 are activated.

Elapsed time spent in DB2 can be divided into three categories:

- Time spent doing work (TCB CPU in DB2)
- Time spent waiting for reasons captured by DB2 (class 3 total wait time)
- Any remaining time after subtracting the two first values from the total elapsed time in DB2

This time is not specifically captured in the DB2 accounting record and is therefore called *not accounted*. It is usually caused by some kind of MVS usage, such as dispatcher waits.

**Tuning Tip:** If this value is high, MVS monitoring tools, such as MAINVIEW for MVS, should be used to determine the cause. However, ignore this value for DDF requester (allied distributed) threads since it includes the time the requester waited for responses from the server and may be skewed by asynchronous activity. The DB2 accounting data at the server location should be used to analyze this time.

ROUTINES (ELAPSED TIME)

Total elapsed time spent executing SQL in stored procedures and user-defined functions, plus the accumulated elapsed time used for executing triggers under an enclave (QWACSPEB + QWACUDEB + QWACTREE) (DB2 6.1 and later).

TCB CPU TIME
CPU time from MVS (QWACEJST - QWACBJST).

This number is the MVS CPU time as reported from all calling sources, such as CICS, IMS, or TSO. Zeros indicate no time is available.

This number represents the MVS CPU time for the application. It is calculated by subtracting the beginning CPU time from the ending CPU time.

TCB CPU IN APPL
CPU time from MVS that is not in DB2 (QWACEJST - QWACBJST - QWACAJST).

This time is the CPU time accumulated when a thread is NOT in DB2. Accounting class 2 must be activated to collect this data.

This time does not include any time spent in a stored procedure, which is collected separately.

TCB CPU IN DB2
DB2 CPU time (QWACAJST).

This time is the CPU time accumulated when a thread is in DB2. Accounting class 2 must be activated to collect this data.

This time does not include any time spent in a stored procedure, which is collected separately.

ROUTINES (TCB CPU TIME)
Accumulated TCB time spent in DB2 processing SQL statements issued by stored procedures and user-defined functions, plus the accumulated TCB time used for executing triggers under an enclave (QWACSPTT + QWACUDTT + QWACTRTE)
(DB2 6.1 and later).

TCB CPU TIME-PRLL TASKS
CPU time spent processing parallel tasks (QWACEJST - QWACBJST).

This number is the MVS CPU time as reported from all calling sources, such as CICS, IMS, or TSO. Zeros indicate no time is available.

This number represents the MVS CPU time for the application. It is calculated by subtracting the beginning CPU time from the ending CPU time.

CLASS 3 TOTAL WAIT
Total of all class 3 elapsed wait times for synchronous events.

Accounting class 3 must be activated to collect this data.

LOCK/LATCH
Wait time for lock/latch (QWACAWTL).

This value indicates the elapsed time the thread waited for transaction locks and latches.

This time reflects the amount of time the thread was suspended because of an incompatible lock or latch mode. Accounting class 3 must be activated to determine this value.

SYNC I/O

Wait time for synchronous I/O (QWACAWTI).

This value indicates the elapsed time spent waiting for all synchronous I/O.

Most of the time included in this value is synchronous read time, but under stress conditions it can reflect synchronous write time (when immediate write threshold is reached). During heavy update activity, it also can reflect wait for synchronous log I/O. Accounting class 3 must be activated to determine this value.

**Note:** Log waits are not included in this value for DB2 6.1 and later.

AVG/SYNC I/O EVENT

Total wait time for synchronous I/O divided by the number of synchronous I/O wait trace events (QWACAWTI/QWACARNE).

**Note:** Log waits are not included in this value for DB2 6.1 and later.

OTHER THREAD READ I/O

Wait time for asynchronous reads (QWACAWTR).

This value is the time spent waiting for reads done under another thread to complete. These waits may be for sequential prefetch, sequential detection, list prefetch, or for synchronous I/O done while executing a sequential prefetch operation.

**Tuning Tip:** Time reflected here does not reflect the duration of the I/O but simply the time the thread waited for completion of the I/O activity. If this number is high, parallel processing may assist in reducing this time. Accounting class 3 must be activated to collect this number.

AVG/ASYNC I/O EVENT

Total wait time for asynchronous reads divided by the number of asynchronous I/O wait trace events (QWACAWTR/QWACARNR).

EXECUTION TASK SWITCH

Wait time for DB2 services (QWACAWTE).

This value is the time spent waiting for special DB2 services. A synchronous execution unit switch is done from the thread to one of the DB2 service tasks.

Service times reported here include

- Commit, abort, or deallocation processing

In DB2 5.1 and below, the following waits are also included in this value:

- Time to recall a page set from HSM (or similar) archive device
- Open/close of a page set
- Update to SYSLGRNG or SYSLGRNX directory page set
- Dataspace manager services
- Define data set (for CREATE operations)
- Extend data set (to take secondary extents)
- Delete data set (for DROP operations)

Accounting class 3 must be activated to obtain this time.

GLOBAL WAIT TIME

Total wait time for data sharing notify messages and global lock contention (QWACAWTG + QWACAWTJ):

- Wait time for data sharing notify messages (QWACAWTG)

  This value is the amount of wait time for sending notify messages to another data sharing member.

  When tables in a data sharing group are CREATed, ALTERed, or DROPped in a data sharing group, intersystem messages must be sent to update the database descriptors (DBDs) in the other members' EDM pools. This is the time spent waiting for this message traffic to occur. This time is collected only when accounting class 3 is activated and data sharing is enabled.

- Wait time for global lock contention (QWACAWTJ)

  This time is the amount of elapsed time caused by an IRLM lock suspension due to global lock contention which requires intersystem communication to resolve.

  This time is collected only when accounting class 3 is activated and data sharing is enabled.

OTHER WAIT TIME

Total of other class 3 wait times (QWACAWTW + QWACALOG or QWAXALOG + QWACAWAR or QWAXAWAR + QWACAWDR or QWAXAWDR + QWACAWCL or QWAXAWCL + QWACAWTP + QWACCAST):

- Wait time for write I/O (QWACAWTW)

  This value is the time spent waiting for an asynchronous write I/O to complete or for synchronous write I/O to be performed for another thread.

  **Tuning Tip:** Time reflected here should be small, since normal checkpoint and deferred write thresholds protect threads from waiting for asynchronous write time. If this number is large, the buffer pool size or deferred write threshold may need to be adjusted. Accounting class 3 must be activated to obtain this time.

- Wait time for log quiesce (QWACALOG or QWAXALOG in DB2 6.1 and later)

  This value is the time spent waiting for an -ARCHIVE LOG MODE(QUIESCE) to take place.

  This process externalizes buffers and switches the active log data set. Time reflected here is the time the thread waited not the time for the command to complete. Accounting class 3 must be activated to obtain this time.

- Wait time for archive read from tape (QWACAWAR or QWAXAWAR in DB2 6.1 and later)

  This is the amount of time spent waiting for reads from an archive tape.

  Time spent waiting for an archive tape is either for the completion of the RECOVER utility or for an application program, which has not committed, to read the archive tape in order to complete backout. Accounting class 3 must be activated to obtain this time.

  **Tuning Tip:** If this time is large, the application program should commit more often. It may also make sense to increase the size and number of active logs.

- Wait time for drain lock (QWACAWDR or QWAXAWDR in DB2 6.1 and later)

  This value is the time spent by a requester of a drain lock, such as a utility or command, waiting to acquire that lock.

  Processes, such as utilities that serialize events to page sets, must acquire a drain lock. This is the time spent waiting for this event. Accounting class 3 must be activated to obtain this time.

- Wait time for claim release (QWACAWCL or QWAXAWCL in DB2 6.1 and later)

  This value is the time spent by the holder of a drain lock waiting for SQL claimers to complete their activity.

  Drainers must wait until the SQL claim count reaches zero. This is the time spent waiting for them to clear. Accounting class 3 must be activated to obtain this time.

- Wait time for page latch contention (QWACAWTP)

  This is the time spent waiting for page latching activities due to lock avoidance. Accounting class 3 must be activated to obtain this time.

  **Tuning Tip:** Page latching can be avoided by scheduling applications when there is a lower likelihood of latch contention.

- Wait time for SPAS TCB scheduling (QWACCAST)

  This time is the elapsed time spent waiting for an available TCB in the Stored Procedures Address Space to become available for scheduling a stored procedure.

  This time should be minimal depending on the workload to the Stored Procedures Address space. It is calculated only if accounting class 3 is active.

## BUFFER ACTIVITY

BPOOL HIT RATIO

Buffer pool hit ratio with prefetch.

This value is calculated as

(Total Pages - Total I/O) / Total Pages x 100.0

where

Total Pages is the number of getpage requests (QBACGET).

Total I/O is the sum of

- Synchronous read I/Os (QBACRIO)
- Asynchronous pages read by sequential prefetch (QBACSIO)

GETPAGES

Number of getpage requests (QBACGET).

This is a count of the number of requests for a data page (successful and unsuccessful) from DB2’s data manager to the buffer manager for normal DB2 processing.

DB2 looks first in a central storage virtual buffer pool, then to an expanded storage hiperpool, if present. If not present, DB2 reads the page from DASD. In parallel query processing, the field counts only the number of successful requests.

**Tuning Tip:** The general rule of thumb for tuning any query is to attempt to find the page in memory if possible, either by maintaining large pools to improve the hit ratio for random reads or by using sequential prefetch, dynamic detection, or list prefetch to improve sequential reads. The higher the ratio of pages found in memory, the less I/O the query costs and the faster the query will run.

SYNCHRONOUS READ I/O

Number of synchronous read I/Os (QBACRIO).

This counter is incremented every time DB2 cannot find a page in memory and must issue a random I/O to DASD to retrieve it.

The overall goal in DB2 tuning is to reduce the number of synchronous I/Os by having needed pages in memory. The thread waits for synchronous I/O activity to be completed before proceeding. This wait time is measured with accounting class 3 and is reflected in the SYNC I/O field in the accounting long report (see page 177).

ASYNC PAGES READ

Number of asynchronous pages read (QBACSIO).

This count is the number of pages read by prefetch operations.

**Tuning Tip:** Generally, unless the application is totally random, the higher the prefetch number, the more likely it is that the data will be in memory when needed. The more successful read-ahead buffering is, the faster the application should perform.

PAGES UPDATED

Number of page updates (QBACSWS).

This counter is incremented every time a page is updated and ready to be externalized to DASD. If the same page is updated twice, for example, the count is incremented by 2. This count includes not only updates to data pages but also to workfile pages, so if a sort is used, this number can be higher than expected.

DB2 keeps committed pages in a buffer pool until a system checkpoint occurs or the deferred write threshold is encountered. DB2 attempts to write data to the page set asynchronously.

The number of pages used in prefetch is governed by the size of the buffer pool. Prefetch can be 8, 16, or 32 pages.

TOTAL PREFETCH

Total number of prefetch requests
(QBACSEQ + QBACLPF + QBACDPF).

This number includes each of these types of prefetch requests:

- Sequential prefetch requests (QBACSEQ)

  This counter is incremented each time DB2 issues a sequential prefetch request. Normally, DB2 will attempt to do this read-ahead buffering by reading up to 32 pages per prefetch request and up to 64 pages per utility prefetch.

  **Tuning Tip:** A buffer pool must be at least 1000 pages to get the full benefit of sequential prefetch. Also, a sufficient number of sequential pages must be allowed in the buffer pool assigned to the accessed table (VPSEQT parameter). Otherwise, the prefetch quantity may be reduced, or prefetch disabled.

- List prefetch requests (QBACLPF)

  This counter is incremented each time DB2 reads index RIDs (Row IDs) in non-matching index scans, multiple index access path selections, or several types of join access paths.

  List prefetch allows DB2 to sort the index RIDs into data row order, thereby enabling sequential prefetch in data sequence, a process which eliminates data page re-reads and allows read-ahead buffering. The RIDs are sorted in the RID pool. List prefetch is not chosen at BIND if DB2 determines that the resultant size of the index RID list would exceed 50% of the RID pool.

  **Tuning Tip:** At execution time, DB2 disables list prefetch if a single index occupies more than 25% of the RID pool or the RID pool storage is exhausted. When list prefetch is disabled, the query becomes a table space scan. This is highly undesirable. If it occurs frequently, change the SQL to reduce the number of RIDs or increase the size of the RID pool.

- Dynamic prefetch requests (QBACDPF)

  Dynamic prefetch is triggered by DB2 sequential detection. If DB2 determines that a cursor is moving sequentially through the data, it automatically enables sequential prefetch.

  This process allows DB2 to improve performance when sequential prefetch was not enabled at BIND but the data is being retrieved sequentially. DB2 examines the data being retrieved and if it determines that the last five of eight pages are sequential, it uses prefetch to request the next set of up to 32 pages until DB2 determines the criteria are no longer met.

RID LIST PROCESSING USED

Number of times RID pool used (QXMIAP).

This counter is incremented every time the RID pool in the database services address space is used. The RID pool is always used to sort index keys by the RBA so that list prefetch (read-ahead buffering) can be enabled against data pages.

**Tuning Tip:** RID pool usage and sizing is critical for effective DB2 performance. The RID pool size is set by the systems installer in DSNZPARM SPRMRDP on the DSNTIPC installation panel.

The DB2 RID pool is used for two purposes:

- To use multiple indexes to access a table

  In this case, the index entries are read, sorted into RID sequence, unioned (for OR processing), or intersected (for AND processing). Then the data from the table is accessed using list prefetch.

- To resolve an answer set from a non-matching index scan or a nested loop join

  Again, the RIDs are sorted into RID sequence and the data is read using list prefetch.

Failure of RID pool storage causes DB2 to revert to table space scan processing. For large tables, this can cause extremely poor performance.

HIPERPOOL SYNC READS

Number of synchronous moves from hiperpool to virtual pool (QBACHRE).

This counter is incremented each time DB2 needs a page in a virtual (central) buffer pool and determines that it is stored in an expanded storage hiperpool.

**Tuning Tip:** DB2 issues a MVPG (movepage) instruction to move the data from expanded to central storage. Updates can occur only in central storage. The process is synchronous in that the thread waits for the activity to complete, but the hiperpool access times are in microseconds (millionths of seconds) compared to a DASD read (thousandths of seconds).

HIPERPOOL ASYNC READS

Number of asynchronous hiperpool pages read (QBACHPG).

This counter reflects the number of pages found in a hiperpool and moved to a virtual buffer pool because of a prefetch operation.

Pages can be moved from hiperpool to virtual pools very quickly; if, as a result of sequential prefetch, the hardware Asynchronous Data Mover Facility (ADMF) is used, the process is very efficient.

COUPLING FAC. PAGES READ

Total number of synchronous coupling facility reads for *invalid buffer* and *buffer not found* (QBGAXD + QBGAMD).

This occurs when a data sharing coupling facility synchronous read was issued for one of these reasons:

- A local virtual buffer pool or hiperpool had a page marked as invalid but the page existed in the group buffer pool
- A page was not found in the local virtual buffer pool or hiperpool but the page existed in the group buffer pool

This occurs normally in a data sharing complex in which data must be shared. The correct data is refreshed from the group buffer pool.

COUPLING FAC. PAGES WRITTEN

Total number of changed and clean pages written to the group buffer pool (QBGASW + QBGAWC):

- Changed pages

  A changed page was written to the group buffer pool (QBGASW).

  Updated pages must be written to the coupling facility when the object is of intersystem interest so other DB2s can refresh their invalidated buffers. Only changed pages are written when GBPCACHE is set to CHANGE.

- Clean pages

  A clean page was written to the group buffer pool (QBGAWC).

  This is done only when GBPCACHE is set to ALL.

  **Tuning Tip:** This can be expensive but does post the group buffer pool with clean pages and may be useful for small tables with high intersystem read interest to reduce contention.

## LOCK ACTIVITY

TIMEOUTS

Number of lock timeouts (QTXATIM).

This number is incremented every time a DB2 thread waits to get a resource longer than the timeout interval. It is specified with the DSNZPARM IRLMRWT on installation panel DSNTIPI. By default, it is 60 seconds. Utilities can be allowed to wait several multiples of IRLMRWT.

**Tuning Tip:** Lock timeouts are usually caused by an application failing to commit in time for the other thread to gain access to data on pages. Often the problem can be resolved by reducing the time between commits and putting updateable statements near to their COMMIT logic. Every time a timeout occurs, DB2 writes the holder and suspender to the MSTR job log. Normally, this number should be as close to zero as possible.

DEADLOCKS

Number of deadlocks (QTXADEA).

This counter is incremented every time DB2 encounters a deadlock situation for which the IRLM must cancel a task involved in a *deadly embrace*.

**Tuning Tip:** Deadlocks are caused by threads requesting access to two resources which can never be resolved. DB2 chooses its victim by selecting the thread which has done the least number of updates. It records the deadlock in the MSTR address job log. In well-tuned systems, this number should be low. The most frequent cause of deadlock situations are ascending key indexes. The Type 2 indexes available in DB2 4.1 can resolve many of these problems because no index locks are taken.

LOCK SUSPENDS

Number of suspends due to lock conflict (QTXASLOC).

This counter is incremented any time a thread has a conflicting lock request, such as an updater requesting exclusive access to a page another thread is using.

**Tuning Tip:** In a multitasking system, suspensions occur in the normal course of the events. If applications are well-tuned, taking frequent commits and holding on to resources for the fewest possible instructions can minimize suspensions. If suspensions cause frequent timeouts, consider row-level locking.

LOCK ESCALATIONS

Total number of lock escalations to shared mode or exclusive mode (QTXALES + QTXALEX):

- Lock escalations to shared mode (QTXALES)

  This counter is incremented every time the number of locks against a single table space exceeds the number set in DSNZPARM NUMLKTS on installation panel DSNTIPJ or the number set in the LOCKMAX clause of the CREATE TABLESPACE statement.

  **Tuning Tip:** This is not a normal situation unless using repeatable read. If it occurs often, consider changing the LOCKSIZE or LOCKMAX to a higher value, or consider binding the plan with cursor stability or uncommitted read (UR).

- Lock escalations to exclusive mode (QTXALEX)

  This counter is incremented every time the number of updateable locks against a single table space exceeds the DSNZPARM NUMLKTS on installation panel DSNTIPJ or in the LOCKMAX clause of the CREATE TABLESPACE statement. It occurs when the LOCKSIZE parameter is specified as ANY and DB2 has escalated the lock owner to an exclusive lock of the entire table.

  **Tuning Tip:** This is an extremely undesirable situation, usually caused by leaving the LOCKSIZE(ANY) default. To resolve this situation, consider changing the parameter to LOCKSIZE (PAGE) or, in special situations, LOCKSIZE(ROW). This causes the offending application to take the -904 unavailable resource error rather than cause general unavailability to the rest of the users. This is almost always caused by application failure to commit in a timely fashion and can be resolved by application code changes as well as by DBA action.

MAXIMUM PAGE LOCKS

Maximum number of page or row locks held (QTXANPL).

This counter represents the highest number of page or row locks held during an execution of a plan.

**Tuning Tip:** This number may not exceed the NUMLKUS (number of locks per user) count in DSNZPARM. If it does, the user gets a -904 resource unavailable message with a 00C90096 reason code. This number is a significant tuning knob which should be observed when migrating an application from one system to another, particularly if the target DB2 has a different number for the maximum locks threshold. The application can free locks by committing resources more often. If LOCKSIZE(ROW) is specified, DB2 holds a lock for each row on the page. Depending on the row size, this can be extremely costly. Consider the cost of row-level locking carefully before implementing.

DATA SHARING SUSPENDS

Total number of global IRLM and XES suspensions and false global suspensions (QTGAIGLO + QTGASGLO + QTGAFLSE):

- Global IRLM suspensions (QTGAIGLO)

  This number is incremented when an incompatible lock is requested on an object (page set, page or row) in a data sharing environment that is being held by another system.

  For a complete picture of global contention, all three global suspension counts must be considered.

  This counter is a measure of intersystem contention. All locks of intrasystem contention are resolved prior to the IRLM checking for intersystem contention. This number is the number of true waits for another thread on a different MVS that has an incompatible lock type to the requester. Global contention requires intersystem communication to resolve the conflict.

- Global XES suspensions (QTGASGLO)

  This counter is a measure of MVS XES global resource contention.

  The MVS XES lock states were in conflict but IRLM lock states were not.

- False global suspensions (QTGAFLSE)

  This counter is incremented every time MVS Cross-System Services detects contention on the resource hash class but not on the resource itself.

  **Tuning Tip:** MVS Cross-System Services uses a hash table for efficient determination of whether a resource is locked. This hash table points to a number of synonyms. When contention exists on the hash class but not on the actual object, this is false contention. The more resources declared to be of intersystem interest, the more chances there are of this condition occurring. The CF lock structure may be too small.

## Accounting Short Report—Packages

This section is produced only if package accounting classes 7/8 are active.

**Column Description**

PROGRAM NAME

Program ID (QPACPKID).

This is the program name for a DBRM or package.

This field correlates to the EXPLAIN program ID.

TYPE

Type of program, either DBRM or package.

OCCURS

Number of occurrences of this program.

SQLSTMT

SQL request count (QPACSQLC).

This is the average number of SQL statements executed from this package or DBRM, including COMMIT or ROLLBACK statements.

This field assists the user in determining which packages are more SQL intensive.

CLASS7 ELAPSE TIME

Elapsed time of a package/DBRM execution (QPACSCT).

This time represents the average elapsed times of all executions of this package or DBRM.

This information can provide more granular information as to which package or DBRM consumed the most resources. Accounting class 7 must be active to gather this data.

CLASS7 TCB TIME

Total CPU time for package/DBRMs (QPACTJST).

This field is the average CPU time for all in-DB2 time spent by all executions of this package or DBRM during the life of the thread.

This number can provide more granularity than plan level accounting to determine where resource overhead is occurring. Accounting class 7 must be active to collect this data.

CLASS8 SUSP TIME

Average of all class 8 elapsed wait times.

SYSEVENT

Average number of waits that contributed to the class 8 elapsed wait time.

## Accounting Short Report—DDF

This section is produced only if distributed work is done.

**Column Description**

REMOTE LOC

Location name of remote site (QLACLOCN).

Location name of a remote site involved in this distributed application, either as a server or a requester.

TR SENT

Number of transactions sent (QLACTRNS).

This counter is incremented at the requester location and reflects the total number of threads requested at the server location. Each one allocates a DBAT thread.

**Tuning Tip:** This data is useful only at the Requester location since DB2 maintains local site autonomy based on the number of active and inactive threads in DSNZPARMs.

TR RECV

Number of transactions received (QLACTRNR).

This counter is incremented at the server location and reflects the number of threads requested by the requester.

**Tuning Tip:** This counter is useful only at the server location since DB2 maintains local site autonomy based on the number of active and inactive threads in DSNZPARMs.

SQL SENT

Number of SQL statements sent to remote server (QLACSQLS).

This counter is incremented at the requester location when an SQL statement is sent.

**Tuning Tip:** This information, along with other message information, can help determine the relative efficiency of the distributed applications and can, in conjunction with other data items, help in the network tuning of distributed applications.

SQL RECV

Number of SQL statements received from requester (QLACSQLR).

This counter is incremented at the server location when an SQL statement is received.

**Tuning Tip:** This information, along with other message information, can help determine the relative efficiency of the distributed applications and can, in conjunction with other data items, help in the network tuning of distributed applications.

ROW SENT

Number of rows sent to requester location (QLACROWS).

This counter is incremented at the server location for each row sent to a remote requester location.

**Tuning Tip:** This number, combined with the number of messages (conversations), can be used in determining whether efficiency techniques such as block fetch are being utilized.

ROW RECV

Number of rows received from the server (QLACROWR).

This counter is incremented at the requester location and reflects the number of rows retrieved from the server.

**Tuning Tip:** From a performance perspective, the ideal is to retrieve as many rows in as few conversational messages as possible. This is accomplished using a feature of DB2 called block fetch. Block fetch is used when a cursor is determined unambiguously to be a read only cursor. The use of the FOR FETCH only clause along with the bind option of CURRENTDATA(NO) can help DB2 achieve this goal. Block fetch is always the most effective with system-directed SQL (three part names without a CONNECT statement, sometimes called DB2 private protocols). In using private protocols, DB2 fits as many rows as possible into a single 32K message. The typical overhead in distributed performance is network bandwidth. If the application can send fewer messages, the distributed application is more efficient. For further information, consult the Performance, Monitoring and Tuning section of the *IBM DATABASE 2 Administration Guide*.

CONV-ALLOC

Number of successful conversation allocations (QLACCNVA).

This value is incremented at the requester location and includes only successful conversation allocations.

CONV-TERM

Number of conversations terminated (QLACCNVT).

This value is incremented at the requester location and includes all conversations terminated.

This value might not be the same as the number allocated because not all conversations are terminated when the accounting record is written.

ELAPSE REMOTE WAIT

Elapsed at requester waiting for DBAT (QLACCPUL).

Elapsed time at the requester location spent waiting for completion of work by a database access agent.

**Tuning Tip:** This value is maintained at the requester location. It includes both DB2 and VTAM processing time as well as network time. It is calculated by accumulating the difference between the store clock value obtained before and after each VTAM request.

# Chapter 8.  Accounting Long Report

This chapter describes the accounting long report, as shown in Figure 15–Figure 19. It has three pages per summary key, followed by pages for package accounting and DDF information, if present.

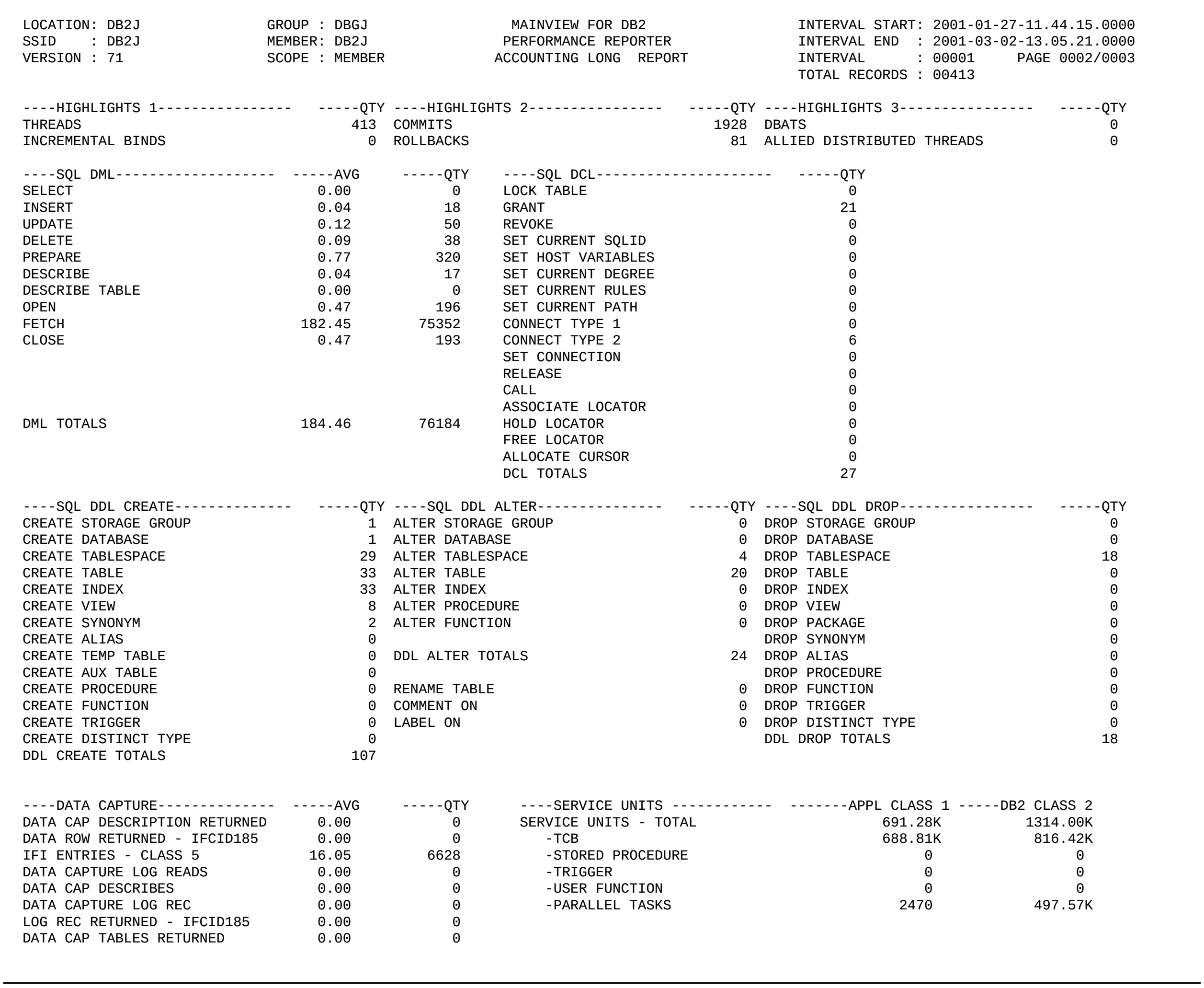
**Note:** Values displayed in the AVG columns are calculated by dividing the total events or times by the number of threads.

```
LOCATION: DB2J               GROUP : DBGJ                 MAINVIEW FOR DB2                  INTERVAL START: 2001-01-27-11.44.15.0000
SSID    : DB2J               MEMBER: DB2J                PERFORMANCE REPORTER               INTERVAL END  : 2001-03-02-13.05.21.0000
VERSION : 71                 SCOPE : MEMBER             ACCOUNTING LONG  REPORT             INTERVAL      : 00001     PAGE 0001/0003
                                                                                            TOTAL RECORDS : 00413

----HIGHLIGHTS 1----------------   -----QTY ----HIGHLIGHTS 2----------------   -----QTY ----HIGHLIGHTS 3----------------   -----QTY
THREADS                                413  COMMITS                              1928  DBATS                                    0
INCREMENTAL BINDS                        0  ROLLBACKS                              81  ALLIED DISTRIBUTED THREADS               0

----AVERAGE TIMES (SECONDS)-----   ----APPL CLASS 1 -----DB2 CLASS 2 ------IFI CLASS 5  ----QUERY PARALLELISM-----------   -----QTY
ELAPSED TIME                             557.9459           13.2541            0.9754   PARALLEL DEGREE < PLANNED                1
    STORED PROC                            0.0000            0.0000               N/A  PARALLEL GRPS FALLBACK-NO BUF            0
    USER FUNCTION                          0.0000            0.0000               N/A  PARALLEL GRPS FALLBACK-CURSOR            1
    TRIGGER                                0.0000            0.0000               N/A  PARALLEL GRPS FALLBACK-NO ESA            0
TCB CPU TIME                               1.1459            1.7076            0.0155   PARALLEL GRPS FALLBACK-ENCLAVE           0
    STORED PROC                            0.0000            0.0000               N/A  SINGLE DB2-COORDINATOR=NO                0
    USER FUNCTION                          0.0000            0.0000               N/A  SINGLE DB2-ISOLATION=RR                  0
    TRIGGER                                0.0000            0.0000               N/A  DB2 MEMBER SKIPPED-NO BUF                3
SRB CPU TIME                               0.2136            0.0069               N/A  PARALLELISM DISABLED BY RLF              0
TCB CPU TIME-PRLL TASKS                    0.0035            1.3599            0.0000   PARALLEL GRPS -PLANNED DEGREE            4
ELAPSED TIME-DATA CAPTURE                     N/A               N/A            0.0000   CPU PARALLEL TASKS CREATED              80
ELAPSED TIME-LOG EXTRACT                      N/A               N/A            0.0000   PARALLEL GROUPS INTENDED                 1
DB2 EXIT/ENTRY EVENTS                         N/A           164.08K               N/A  PARALLEL GROUPS EXECUTED                 6
CLASS 1 NOT NULL COUNT                        413               413                52   MAX DEGREE OF PARALLELISM               21
                                                                                        MAX MEMBERS PARTICIPATED           131.07K
                                                                                        REFORM - CONFIG CHANGE                   0
                                                                                        REFORM - NO BUFFER                       1

----CLASS 3 SUSPENSION---------- ---------AV.TIME -AV.EVENT             ----LOCKING ACTIVITY----------  -----AVG     -----QTY
LOCK/LATCH                                0.5774    107.51              TIMEOUTS                           0.04           15
SYNC I/O                                  1.0556     81.76              DEADLOCKS                          0.01            3
   DATABASE                               0.7636     34.96              LOCK SUSPENDS                      0.06           25
   LOG WRITE                              0.2920     46.80              OTHER SUSPENDS                    32.46        13407
OTHER THREAD READ I/O                     0.6525     53.87              LATCH SUSPENDS                     0.05           22
OTHER THREAD WRITE I/O                    0.0035      0.16              MAXIMUM PAGE LOCKS                 0.00           29
EXECUTION TASK SWITCH                     4.5433     30.80              LOCK REQUESTS                    453.00       187.09K
   COMMIT/UPDATE                          0.0886      3.94              LOCK QUERY REQUESTS                2.23          920
   OPEN/CLOSE                             3.6463      6.78              LOCK CHANGE REQUESTS               9.72         4014
   SYSLGRNG REC                           0.2703      6.26              OTHER IRLM REQUESTS                0.00            0
   DEFINE/EXTEND/DELETE                   0.3047      1.31              UNLOCK REQUESTS                  360.22       148.77K
   OTHER                                  0.2330     12.50              LOCK ESCALATIONS TO SHARED         0.00            0
ARCHIVE LOG QUIESCE                       0.0000      0.00              LOCK ESCALATIONS TO EXCLUSIVE      0.00            0
ARCHIVE LOG READ                          0.0000      0.00              LOCK TOTALS                      857.77       354.26K
DRAIN LOCK                                0.0000      0.00
CLAIM RELEASE                             0.0000      0.00              ----OPTIMIZATION -------------  -----AVG     -----QTY
PAGE LATCH CONTENTION                     0.0000      0.00              REOPTIMIZATION                     0.00            0
STORED PROCEDURE                          0.0000      0.00              PREPARE - SATISFIED                0.26          107
USER DEFINED FUNCTION                     0.0000      0.00                      - FAILED SEARCH            0.56          233
NOTIFY MESSAGES                           0.1405     32.01                      - IMPLICIT                 0.00            0
GLOBAL CONTENTION                         1.4743      1.62                      - FROM CACHE               0.00            0
CLASS 3 TOTAL WAIT                        8.4476    307.72                      - DISCARDED MAX            0.00            0
CLASS 3 NOT NULL COUNT                       387                                - DROP/ALTER/REVOKE        0.03           12

----DRAIN / CLAIM-------------  -----AVG     -----QTY                   ----RID LIST------------------  -----AVG     -----QTY
CLAIM REQUESTS                    29.03        11988                    RID LIST PROCESSING USED           0.00            0
FAILED CLAIM REQUESTS              0.00            0                    RID FAILURES - OVER MAX            0.00            0
DRAIN REQUESTS                     0.56          230                    RID FAILURES - STORAGE             0.00            0
FAILED DRAIN REQUESTS              0.00            0
```

Figure 15.  Accounting Long Report—Page 1

For a complete description of each section, see

- “HIGHLIGHTS” on page 130
- “AVERAGE TIMES (SECONDS)” on page 131
- “QUERY PARALLELISM” on page 136
- “CLASS 3 SUSPENSION” on page 139
- “LOCKING ACTIVITY” on page 143
- “DRAIN / CLAIM” on page 146
- “DATA CAPTURE” on page 159
- “RID LIST” on page 148



```
LOCATION: DB2J               GROUP : DBGJ                 MAINVIEW FOR DB2                  INTERVAL START: 2001-01-27-11.44.15.0000
SSID    : DB2J               MEMBER: DB2J                PERFORMANCE REPORTER               INTERVAL END  : 2001-03-02-13.05.21.0000
VERSION : 71                 SCOPE : MEMBER             ACCOUNTING LONG  REPORT             INTERVAL      : 00001     PAGE 0002/0003
                                                                                            TOTAL RECORDS : 00413

----HIGHLIGHTS 1----------------   -----QTY ----HIGHLIGHTS 2----------------   -----QTY ----HIGHLIGHTS 3----------------   -----QTY
THREADS                                413  COMMITS                               1928  DBATS                                    0
INCREMENTAL BINDS                        0  ROLLBACKS                               81  ALLIED DISTRIBUTED THREADS               0

----SQL DML-------------------  -----AVG     -----QTY    ----SQL DCL---------------------   -----QTY
SELECT                             0.00            0     LOCK TABLE                               0
INSERT                             0.04           18     GRANT                                   21
UPDATE                             0.12           50     REVOKE                                   0
DELETE                             0.09           38     SET CURRENT SQLID                        0
PREPARE                            0.77          320     SET HOST VARIABLES                       0
DESCRIBE                           0.04           17     SET CURRENT DEGREE                       0
DESCRIBE TABLE                     0.00            0     SET CURRENT RULES                        0
OPEN                               0.47          196     SET CURRENT PATH                         0
FETCH                            182.45        75352     CONNECT TYPE 1                           0
CLOSE                              0.47          193     CONNECT TYPE 2                           6
                                                         SET CONNECTION                           0
                                                         RELEASE                                  0
                                                         CALL                                     0
                                                         ASSOCIATE LOCATOR                        0
DML TOTALS                       184.46        76184     HOLD LOCATOR                             0
                                                         FREE LOCATOR                             0
                                                         ALLOCATE CURSOR                          0
                                                         DCL TOTALS                              27

----SQL DDL CREATE--------------   -----QTY ----SQL DDL ALTER---------------   -----QTY ----SQL DDL DROP----------------   -----QTY
CREATE STORAGE GROUP                     1  ALTER STORAGE GROUP                      0  DROP STORAGE GROUP                       0
CREATE DATABASE                          1  ALTER DATABASE                           0  DROP DATABASE                            0
CREATE TABLESPACE                       29  ALTER TABLESPACE                         4  DROP TABLESPACE                         18
CREATE TABLE                            33  ALTER TABLE                             20  DROP TABLE                               0
CREATE INDEX                            33  ALTER INDEX                              0  DROP INDEX                               0
CREATE VIEW                              8  ALTER PROCEDURE                          0  DROP VIEW                                0
CREATE SYNONYM                           2  ALTER FUNCTION                           0  DROP PACKAGE                             0
CREATE ALIAS                             0                                              DROP SYNONYM                             0
CREATE TEMP TABLE                        0  DDL ALTER TOTALS                        24  DROP ALIAS                               0
CREATE AUX TABLE                         0                                              DROP PROCEDURE                           0
CREATE PROCEDURE                         0  RENAME TABLE                             0  DROP FUNCTION                            0
CREATE FUNCTION                          0  COMMENT ON                               0  DROP TRIGGER                             0
CREATE TRIGGER                           0  LABEL ON                                 0  DROP DISTINCT TYPE                       0
CREATE DISTINCT TYPE                     0                                              DDL DROP TOTALS                         18
DDL CREATE TOTALS                      107


----DATA CAPTURE--------------  -----AVG     -----QTY      ----SERVICE UNITS -----------  -------APPL CLASS 1 -----DB2 CLASS 2
DATA CAP DESCRIPTION RETURNED      0.00            0       SERVICE UNITS - TOTAL                     691.28K         1314.00K
DATA ROW RETURNED - IFCID185       0.00            0          -TCB                                   688.81K          816.42K
IFI ENTRIES - CLASS 5             16.05         6628          -STORED PROCEDURE                            0                0
DATA CAPTURE LOG READS             0.00            0          -TRIGGER                                     0                0
DATA CAP DESCRIBES                 0.00            0          -USER FUNCTION                               0                0
DATA CAPTURE LOG REC               0.00            0          -PARALLEL TASKS                           2470          497.57K
LOG REC RETURNED - IFCID185        0.00            0
DATA CAP TABLES RETURNED           0.00            0
```

Figure 16. Accounting Long Report—Page 2

For a complete description of each section, see

- "SQL DML" on page 149
- "SQL DCL" on page 151
- "SQL DDL CREATE" on page 154
- "SQL DDL ALTER" on page 156
- "SQL DDL DROP" on page 157
- "DATA CAPTURE" on page 159
- "SERVICE UNITS" on page 160

```
LOCATION: DB2J               GROUP : DBGJ                 MAINVIEW FOR DB2                  INTERVAL START: 2001-01-27-11.44.15.0000
SSID    : DB2J               MEMBER: DB2J                PERFORMANCE REPORTER               INTERVAL END  : 2001-03-02-13.05.21.0000
VERSION : 71                 SCOPE : MEMBER             ACCOUNTING LONG  REPORT             INTERVAL      : 00001     PAGE 0003/0003
                                                                                            TOTAL RECORDS : 00413

----HIGHLIGHTS 1----------------   -----QTY ----HIGHLIGHTS 2----------------   -----QTY ----HIGHLIGHTS 3----------------   -----QTY
THREADS                                 413 COMMITS                                1928 DBATS                                     0
INCREMENTAL BINDS                         0 ROLLBACKS                                81 ALLIED DISTRIBUTED THREADS                0

----BUFFER  POOL -------------  -----AVG     -----QTY      ---TERMINATION REASON-----------   -----QTY
BPOOL HIT RATIO(%)                                ***      NORMAL    - NEW USER                     0
GETPAGES                           1.88K      775.12K               - RESIGNON                      0
SYNCHRONOUS READ I/O              34.07        14071                -  DBAT INACTIVE                0
GETPAGES / READIO                  0.13           55                -  DEALLOCATION               331
SYNCHRONOUS WRITE I/O              0.36          147                -  RRSAF COMMIT                 0
ASYNC PAGES READ                   1.96K      810.68K               -  APPL PROG END               26
SEQUENTIAL PREFETCH               56.71        23423                - IFI READS                     0
LIST PREFETCH                      0.00            0       ABNORMAL - APPL PROG ABEND              19
DYNAMIC PREFETCH                   0.19           78                - END-OF-MEMORY                 1
PAGES UPDATED                    119.52        49360                - RESOLVE INDOUBT              36
HIPERPOOL ASYNC READS              0.00            0                - CANCEL/FORCE DB2              0
HIPERPOOL SYNC READS               0.00            0       INDOUBT - APPL PROG END                  0
HIPERPOOL READ FAIL                0.00            0                - APPL PROG ABEND               0
HIPERPOOL WRITES                   0.00            0                - END-OF-MEMORY                 0
HIPERPOOL WRITES FAIL              0.00            0                - CANCEL/FORCE DB2              0


----GLOBAL BUFFER  POOL ------  -----AVG     -----QTY      ----ROUTINES------------------  -----AVG     -----QTY
SYNC RD INV BUFF -WITH DATA        0.03           12       STORED PROCS -CALL STATEMENTS      0.00            0
          - NO DATA                0.01            3                    -ABENDS               0.00            0
SYNC RD NOT FOUND-WITH DATA        0.02            7                    -TIMEOUTS             0.00            0
          - NO DATA                0.09           37                    -REJECTS              0.00            0
PAGES WRITTEN     -CHANGED         1.19          492       USER DEF FUNC-EXECUTED             0.00            0
                  -CLEAN           0.00            0                    -ABENDS               0.00            0
UNREGISTER PAGES                   0.44          183                    -TIMEOUTS             0.00            0
EXPLICIT X-INVALID                 0.00            0                    -REJECTS              0.00            0
WRITE TO SECONDARY                 0.00            0       TRIGGER-STATEMENT ACTIVATIONS      0.00            0
                                                                  -ROW       ACTIVATIONS      0.00            0
                                                                  -ERRORS                     0.00            0
                                                           MAX NESTED SQL CASCADING           0.00            0

---GLOBAL LOCKING ACTIVITY----  -----AVG     -----QTY      ----MISCELLANEOUS ------------  -----AVG     -----QTY
PLOCK REQUESTS-LOCK                9.81         4052       ROWID - DIRECT ACCESS              0.00            0
              -CHANGE              1.30          537             - INDEX USED                 0.00            0
              -UNLOCK              1.11          458             - TS SCAN USED               0.00            0
XES REQUESTS  -LOCK              143.96        59454       MAX LOB STORAGE                    0.00            0
              -CHANGE              1.54          638       LOG RECS WRITTEN                 133.68        55211
              -UNLOCK             78.22        32303       LOG BYTES WRITTEN                  0.65          269
SUSPENDS      -IRLM                0.72          298
              -XES                 0.24           98
              -FALSE               0.68          279
INCOMPATIBLE RETAINED LOCK         0.00            0
NOTIFY MESSAGES SENT              47.44        19593
GLOBAL LOCK CONTENTION(%)                          0
FALSE CONTENTION(%)                               41
```

Figure 17. Accounting Long Report—Page 3

For a complete description of each section, see

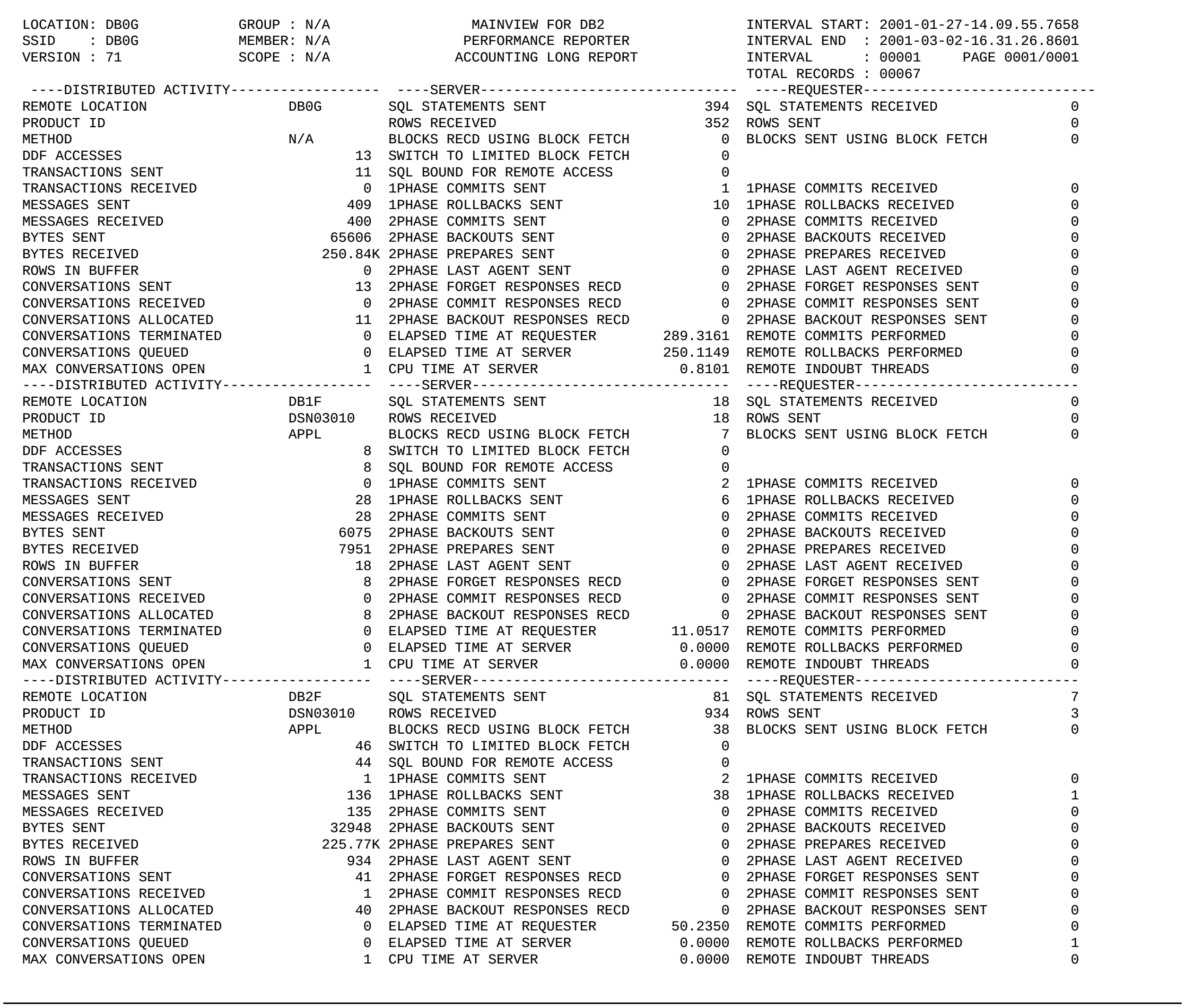
- “BUFFER POOL” on page 161
- “GLOBAL BUFFER POOL” on page 165
- “GLOBAL LOCKING ACTIVITY” on page 167
- “TERMINATION REASON” on page 170
- “ROUTINES” on page 172
- “MISCELLANEOUS” on page 174

```
LOCATION: DB2J               GROUP : DBGJ                 MAINVIEW FOR DB2                  INTERVAL START: 2001-01-27-11.56.59.2758
SSID    : DB2J               MEMBER: DB2J                PERFORMANCE REPORTER               INTERVAL END  : 2001-03-02-13.05.21.6690
VERSION : 71                 SCOPE : MEMBER             ACCOUNTING LONG  REPORT             INTERVAL      : 00001     PAGE 0001/....
                                                                                            TOTAL RECORDS : .....


--- PACKAGE / DBRM ACTIVITY ----------------                         ---- PKG./ DBRM WAIT TIMES ---- ----------AV.TIME -AV.EVENT
TYPE (PKG, DBRM, BOTH)          DBRM                                 LOCK/LATCH                                0.0000      1.00
PACKAGE COLLECTION ID                                                SYNC I/O                                  0.4685     50.50
LOCATION NAME                                                        OTHER THREAD READ I/O                     0.7104     49.00
PROGRAM NAME                    DPSPURGD                             OTHER THREAD WRITE I/O                    0.1070      4.50
                                                                     EXECUTION TASK SWITCH                   204.8912     70.50
OCCURRENCES                                      2                    ARCHIVE LOG QUIESCE                       0.0000      0.00
SQL STATEMENTS COUNT                            40                    ARCHIVE LOG READ                          0.0000      0.00
DB2 EXIT/ENTRY EVENTS                          120                    DRAIN LOCK                                0.0000      0.00
TOTAL ELAPSED EXECUTION TIME              207.0102                    CLAIM RELEASE                             0.0000      0.00
TOTAL TCB CPU TIME IN PACKAGE               0.5441                    PAGE LATCH CONTENTION                     0.0000      0.00
      CPU IN PARALLEL TASKS                 0.0000                    STORED PROCEDURE                          0.0000      0.00
TOTAL WAIT TIME                           206.1772                    NOTIFY MESSAGES                           0.0000      0.00
CLASS 7 NOT NULL COUNT                           2                    GLOBAL CONTENTION                         0.0000      0.00
SERVICE UNITS - TOTAL                         2268                    CLASS 8 SUSPENSION TIME                 206.1772      0.00
      -PARALLEL TASKS                            0                    CLASS 8 NOT NULL COUNT                         2

--- PACKAGE / DBRM ACTIVITY ----------------                         ---- PKG./ DBRM WAIT TIMES ---- ----------AV.TIME -AV.EVENT
TYPE (PKG, DBRM, BOTH)          DBRM                                 LOCK/LATCH                                0.0000      0.11
PACKAGE COLLECTION ID                                                SYNC I/O                                  0.0859      9.44
LOCATION NAME                                                        OTHER THREAD READ I/O                     0.1359     10.22
PROGRAM NAME                    DPSQLDAD                             OTHER THREAD WRITE I/O                    0.0000      0.00
                                                                     EXECUTION TASK SWITCH                     0.4941      1.44
OCCURRENCES                                      9                    ARCHIVE LOG QUIESCE                       0.0000      0.00
SQL STATEMENTS COUNT                          2372                    ARCHIVE LOG READ                          0.0000      0.00
DB2 EXIT/ENTRY EVENTS                         4706                    DRAIN LOCK                                0.0000      0.00
TOTAL ELAPSED EXECUTION TIME               26.1583                    CLAIM RELEASE                             0.0000      0.00
TOTAL TCB CPU TIME IN PACKAGE               0.5320                    PAGE LATCH CONTENTION                     0.0000      0.00
      CPU IN PARALLEL TASKS                 0.0000                    STORED PROCEDURE                          0.0000      0.00
TOTAL WAIT TIME                             0.7160                    NOTIFY MESSAGES                           0.0000      0.00
CLASS 7 NOT NULL COUNT                           9                    GLOBAL CONTENTION                         0.0000      0.00
SERVICE UNITS - TOTAL                         7768                    CLASS 8 SUSPENSION TIME                   0.7160      0.00
      -PARALLEL TASKS                            0                    CLASS 8 NOT NULL COUNT                         9

--- PACKAGE / DBRM ACTIVITY ----------------                         ---- PKG./ DBRM WAIT TIMES ---- ----------AV.TIME -AV.EVENT
TYPE (PKG, DBRM, BOTH)          PKG                                  LOCK/LATCH                                0.4392      0.80
PACKAGE COLLECTION ID           SMRTEP2A                             SYNC I/O                                 19.4662    524.87
LOCATION NAME                   DB2J                                 OTHER THREAD READ I/O                     1.6719    134.33
PROGRAM NAME                    DSN@EP2L                             OTHER THREAD WRITE I/O                    0.0000      0.00
                                                                     EXECUTION TASK SWITCH                    56.6636    293.20
OCCURRENCES                                     15                    ARCHIVE LOG QUIESCE                       0.0000      0.00
SQL STATEMENTS COUNT                            51                    ARCHIVE LOG READ                          0.0000      0.00
DB2 EXIT/ENTRY EVENTS                          140                    DRAIN LOCK                                0.0001      0.07
TOTAL ELAPSED EXECUTION TIME               85.7050                    CLAIM RELEASE                             0.0000      0.00
TOTAL TCB CPU TIME IN PACKAGE               1.4418                    PAGE LATCH CONTENTION                     0.0000      0.00
      CPU IN PARALLEL TASKS                 0.0173                    STORED PROCEDURE                          0.0000      0.00
TOTAL WAIT TIME                            78.9571                    NOTIFY MESSAGES                           0.0000      0.00
CLASS 7 NOT NULL COUNT                          15                    GLOBAL CONTENTION                         0.0000     23.93
SERVICE UNITS - TOTAL                        19579                    CLASS 8 SUSPENSION TIME                  78.9571      0.00
      -PARALLEL TASKS                          230                    CLASS 8 NOT NULL COUNT                        15
```

Figure 18. Accounting Long Report—Packages

For a complete description of each section, see

- “PACKAGE / DBRM ACTIVITY” on page 175
- “PKG./ DBRM WAIT TIMES” on page 177



```
LOCATION: DB0G           GROUP : N/A              MAINVIEW FOR DB2                INTERVAL START: 2001-01-27-14.09.55.7658
SSID    : DB0G           MEMBER: N/A            PERFORMANCE REPORTER              INTERVAL END  : 2001-03-02-16.31.26.8601
VERSION : 71             SCOPE : N/A            ACCOUNTING LONG REPORT            INTERVAL      : 00001     PAGE 0001/0001
                                                                                  TOTAL RECORDS : 00067
 ----DISTRIBUTED ACTIVITY------------------  ----SERVER------------------------------  ----REQUESTER---------------------------
REMOTE LOCATION                DB0G       SQL STATEMENTS SENT                 394  SQL STATEMENTS RECEIVED                0
PRODUCT ID                                ROWS RECEIVED                       352  ROWS SENT                              0
METHOD                         N/A        BLOCKS RECD USING BLOCK FETCH         0  BLOCKS SENT USING BLOCK FETCH          0
DDF ACCESSES                           13 SWITCH TO LIMITED BLOCK FETCH         0
TRANSACTIONS SENT                      11 SQL BOUND FOR REMOTE ACCESS           0
TRANSACTIONS RECEIVED                   0 1PHASE COMMITS SENT                   1  1PHASE COMMITS RECEIVED                0
MESSAGES SENT                         409 1PHASE ROLLBACKS SENT                10  1PHASE ROLLBACKS RECEIVED              0
MESSAGES RECEIVED                     400 2PHASE COMMITS SENT                   0  2PHASE COMMITS RECEIVED                0
BYTES SENT                          65606 2PHASE BACKOUTS SENT                  0  2PHASE BACKOUTS RECEIVED               0
BYTES RECEIVED                    250.84K 2PHASE PREPARES SENT                  0  2PHASE PREPARES RECEIVED               0
ROWS IN BUFFER                          0 2PHASE LAST AGENT SENT                0  2PHASE LAST AGENT RECEIVED             0
CONVERSATIONS SENT                     13 2PHASE FORGET RESPONSES RECD          0  2PHASE FORGET RESPONSES SENT           0
CONVERSATIONS RECEIVED                  0 2PHASE COMMIT RESPONSES RECD          0  2PHASE COMMIT RESPONSES SENT           0
CONVERSATIONS ALLOCATED                11 2PHASE BACKOUT RESPONSES RECD         0  2PHASE BACKOUT RESPONSES SENT          0
CONVERSATIONS TERMINATED                0 ELAPSED TIME AT REQUESTER      289.3161  REMOTE COMMITS PERFORMED               0
CONVERSATIONS QUEUED                    0 ELAPSED TIME AT SERVER         250.1149  REMOTE ROLLBACKS PERFORMED             0
MAX CONVERSATIONS OPEN                  1 CPU TIME AT SERVER               0.8101  REMOTE INDOUBT THREADS                 0
----DISTRIBUTED ACTIVITY------------------  ----SERVER------------------------------  ----REQUESTER--------------------------
REMOTE LOCATION                DB1F       SQL STATEMENTS SENT                  18  SQL STATEMENTS RECEIVED                0
PRODUCT ID                     DSN03010   ROWS RECEIVED                        18  ROWS SENT                              0
METHOD                         APPL       BLOCKS RECD USING BLOCK FETCH         7  BLOCKS SENT USING BLOCK FETCH          0
DDF ACCESSES                            8 SWITCH TO LIMITED BLOCK FETCH         0
TRANSACTIONS SENT                       8 SQL BOUND FOR REMOTE ACCESS           0
TRANSACTIONS RECEIVED                   0 1PHASE COMMITS SENT                   2  1PHASE COMMITS RECEIVED                0
MESSAGES SENT                          28 1PHASE ROLLBACKS SENT                 6  1PHASE ROLLBACKS RECEIVED              0
MESSAGES RECEIVED                      28 2PHASE COMMITS SENT                   0  2PHASE COMMITS RECEIVED                0
BYTES SENT                           6075 2PHASE BACKOUTS SENT                  0  2PHASE BACKOUTS RECEIVED               0
BYTES RECEIVED                       7951 2PHASE PREPARES SENT                  0  2PHASE PREPARES RECEIVED               0
ROWS IN BUFFER                         18 2PHASE LAST AGENT SENT                0  2PHASE LAST AGENT RECEIVED             0
CONVERSATIONS SENT                      8 2PHASE FORGET RESPONSES RECD          0  2PHASE FORGET RESPONSES SENT           0
CONVERSATIONS RECEIVED                  0 2PHASE COMMIT RESPONSES RECD          0  2PHASE COMMIT RESPONSES SENT           0
CONVERSATIONS ALLOCATED                 8 2PHASE BACKOUT RESPONSES RECD         0  2PHASE BACKOUT RESPONSES SENT          0
CONVERSATIONS TERMINATED                0 ELAPSED TIME AT REQUESTER       11.0517  REMOTE COMMITS PERFORMED               0
CONVERSATIONS QUEUED                    0 ELAPSED TIME AT SERVER           0.0000  REMOTE ROLLBACKS PERFORMED             0
MAX CONVERSATIONS OPEN                  1 CPU TIME AT SERVER               0.0000  REMOTE INDOUBT THREADS                 0
----DISTRIBUTED ACTIVITY------------------  ----SERVER------------------------------  ----REQUESTER--------------------------
REMOTE LOCATION                DB2F       SQL STATEMENTS SENT                  81  SQL STATEMENTS RECEIVED                7
PRODUCT ID                     DSN03010   ROWS RECEIVED                       934  ROWS SENT                              3
METHOD                         APPL       BLOCKS RECD USING BLOCK FETCH        38  BLOCKS SENT USING BLOCK FETCH          0
DDF ACCESSES                           46 SWITCH TO LIMITED BLOCK FETCH         0
TRANSACTIONS SENT                      44 SQL BOUND FOR REMOTE ACCESS           0
TRANSACTIONS RECEIVED                   1 1PHASE COMMITS SENT                   2  1PHASE COMMITS RECEIVED                0
MESSAGES SENT                         136 1PHASE ROLLBACKS SENT                38  1PHASE ROLLBACKS RECEIVED              1
MESSAGES RECEIVED                     135 2PHASE COMMITS SENT                   0  2PHASE COMMITS RECEIVED                0
BYTES SENT                          32948 2PHASE BACKOUTS SENT                  0  2PHASE BACKOUTS RECEIVED               0
BYTES RECEIVED                    225.77K 2PHASE PREPARES SENT                  0  2PHASE PREPARES RECEIVED               0
ROWS IN BUFFER                        934 2PHASE LAST AGENT SENT                0  2PHASE LAST AGENT RECEIVED             0
CONVERSATIONS SENT                     41 2PHASE FORGET RESPONSES RECD          0  2PHASE FORGET RESPONSES SENT           0
CONVERSATIONS RECEIVED                  1 2PHASE COMMIT RESPONSES RECD          0  2PHASE COMMIT RESPONSES SENT           0
CONVERSATIONS ALLOCATED                40 2PHASE BACKOUT RESPONSES RECD         0  2PHASE BACKOUT RESPONSES SENT          0
CONVERSATIONS TERMINATED                0 ELAPSED TIME AT REQUESTER       50.2350  REMOTE COMMITS PERFORMED               0
CONVERSATIONS QUEUED                    0 ELAPSED TIME AT SERVER           0.0000  REMOTE ROLLBACKS PERFORMED             1
MAX CONVERSATIONS OPEN                  1 CPU TIME AT SERVER               0.0000  REMOTE INDOUBT THREADS                 0
```

Figure 19.  Accounting Long Report—DDF

For a complete description of each section, see

- “DISTRIBUTED ACTIVITY” on page 180
- “SERVER” on page 183
- “REQUESTER” on page 187

# Accounting Long Report—Page 1

The fields in this section are organized in the following categories:

## HIGHLIGHTS

THREADS
Number of thread executions.

INCREMENTAL BINDS
Number of incremental BINDs performed (QXINCRB).

This counter is incremented every time a plan is run that was bound with the VALIDATE(RUN) option.

**Tuning Tip:** It is generally undesirable to bind a plan with the VALIDATE(RUN) option since all SQL statements must be rechecked for syntax, authority, and access path every time the plan is executed. VALIDATE(RUN) is required if the program is going to CREATE TABLES (for example, work tables) during the execution, or if testing is required on a piece of code for which the objects do not yet exist. Otherwise, VALIDATE(RUN) should be avoided, as the cost is nearly that of dynamic SQL.

COMMITS
Number of commits processed (QWACCOMM).

This is a count of the number of commits taken for the thread. It includes both phase 2 and single-phase commits.

**Tuning Tip:** This is a useful number to observe while tracking a long job with a known commit frequency. The commits counted are successful.

ROLLBACKS
Number of rollbacks processed (QWACABRT).

This number represents the total number of rollbacks including application abends, application explicit ROLLBACK, application deadlock, application CANCELed by operator or abended due to DB2 resource shortage.

DBATS
Number of Database Access Threads executed.

ALLIED DISTRIBUTED THREADS
Number of allied distributed threads executed.

## AVERAGE TIMES (SECONDS)

ELAPSED TIME

APPL CLASS 1
Elapsed time from the first connect to DB2 to thread termination (QWACESC - QWACBSC).

The time a thread was inactive waiting for work is included in this value.

This value is not accurate for threads that do not terminate, such as a CICS conversational transaction or an IMS WFI BMP.

DB2 CLASS 2
Elapsed time in DB2 (QWACASC).

This value is collected when accounting class 2 is activated.

In-DB2 time is elapsed time while performing DB2 work. The DB2 time is a percentage of the total elapsed time.

IFI CLASS 5
Elapsed time processing IFI calls (QIFAAIET).

This is the elapsed time processing Instrumentation Facility Interface READA and READS requests to process DB2 trace data.

This information is only gathered if accounting class 5 is active.

ELAPSED TIME-STORED PROC

APPL CLASS 1
Total elapsed time spent in stored procedures, including time spent executing SQL (QWACSPEA).

(DB2 6.1 and later)

DB2 CLASS 2
Total elapsed time spent executing SQL in stored procedures (QWACSPEB).

(DB2 6.1 and later)

IFI CLASS 5
N/A

ELAPSED TIME-USER FUNCTION

APPL CLASS 1
Total elapsed time spent in user-defined functions, including time spent executing SQL (QWACUDEA).

This time is collected with accounting class 1 active.

(DB2 6.1 and later)

DB2 CLASS 2
Total elapsed time spent for user-defined functions to execute SQL (QWACUDEB).

This number is calculated only if accounting class 3 is active.

(DB2 6.1 and later)

IFI CLASS 5
N/A

ELAPSED TIME-TRIGGER

APPL CLASS 1
Accumulated elapsed time used while executing under the control of triggers (QWACTRET).

(DB2 6.1 and later)

DB2 CLASS 2
Accumulated elapsed time used for executing triggers under an enclave (QWACTREE).

(DB2 6.1 and later)

IFI CLASS 5
N/A

TCB CPU TIME

APPL CLASS 1
CPU time from MVS (QWACEJST - QWACBJST).

This number is the MVS CPU time as reported from all calling sources, such as CICS, IMS or TSO. It includes both the main task and all parallel tasks. Zeros indicate no time is available.

This number represents the MVS CPU time for the application. It is calculated by subtracting the beginning CPU time from the ending CPU time.

DB2 CLASS 2
DB2 CPU time (QWACAJST).

This time is the CPU time accumulated when a thread is in DB2. Accounting class 2 must be activated to collect this data.

This time does not include any time spent in a stored procedure, which is collected separately.

IFI CLASS 5
CPU time processing IFI calls (QIFAAITT).

This is the CPU time consumed processing Instrumentation Facility READA and READS requests to process DB2 trace data.

This information is only gathered if accounting class 5 is active.

TCB CPU TIME-STORED PROC

APPL CLASS 1
TCB time spent processing SQL stored procedure CALL statements (QWACSPCP).

This time reflects the TCB time for SQL CALL statements which reference stored procedures in the stored procedures address space or a WLM address space.

This time is collected with accounting class 1 active.

DB2 CLASS 2
TCB time spent in DB2 processing SQL statements issued by stored procedures (QWACSPTT).

This time is the TCB time spent in DB2 processing SQL statements issued by stored procedures.

This number is calculated only if accounting class 2 is active.

IFI CLASS 5
N/A

TCB CPU TIME-USER FUNCTION

APPL CLASS 1
Accumulated TCB time used to satisfy user-defined function requests processed in a DB2 stored procedures address space or WLM established address space (QWACUDCP).

This time is collected with accounting class 1 active.

(DB2 6.1 and later)

DB2 CLASS 2
Accumulated TCB time in DB2 for processing SQL statements issued by user-defined functions (QWACUDTT).

This is the time not included in QWACUDCP.

This number is calculated only if accounting class 2 is active.

(DB2 6.1 and later)

IFI CLASS 5
N/A

TCB CPU TIME-TRIGGER

APPL CLASS 1
Accumulated TCB time used while executing under the control of triggers (QWACTRTT).

(DB2 6.1 and later)

DB2 CLASS 2
Accumulated TCB time used for executing triggers under an enclave (QWACTRTE).

(DB2 6.1 and later)

IFI CLASS 5
N/A

SRB CPU TIME

APPL CLASS 1

SRB time from first connect to DB2 to thread termination or reuse (QWACESRB - QWACBSRB).

This value is the SRB time from the calling address space when the thread is deallocated or reused. Some address spaces, such as CICS and the distributed data facility, that create multiple threads reflect the SRB time for the entire address space instead of for this particular thread.

DB2 CLASS 2

SRB time while in DB2 (QWACASRB).

This time is the accumulated SRB time from the calling address space spent while a thread is in DB2. Accounting class 2 must be activated to collect this data.

Address spaces with multiple connections to DB2, such as CICS or DDF, reflect the SRB time for the entire address space instead of for this particular thread.

IFI CLASS 5

N/A

TCB CPU TIME-PRLL TASKS

APPL CLASS 1

CPU time spent processing parallel tasks (QWACEJST - QWACBJST).

This number is the MVS CPU time as reported from all calling sources, such as CICS, IMS, or TSO. It does not include the main task. Zeros indicate no time is available.

This number represents the MVS CPU time for the application. It is calculated by subtracting the beginning CPU time from the ending CPU time.

DB2 CLASS 2

DB2 CPU time (QWACAJST).

This time is the CPU time accumulated processing parallel tasks when a thread is in DB2. Accounting class 2 must be activated to collect this data.

This time does not include any time spent in a stored procedure, which is collected separately.

IFI CLASS 5

CPU time processing IFI calls for parallel tasks (QIFAAITT).

This is the CPU time consumed processing Instrumentation Facility READA and READS requests to process DB2 trace data.

This information is only gathered if accounting class 5 is active.

ELAPSED TIME-DATA CAPTURE

APPL CLASS 1

N/A

DB2 CLASS 2

N/A

IFI CLASS 5

Elapsed time processing data capture describes (QIFAAMBT).

This is the elapsed time processing data capture describe information during “IFI READS” calls for IFCID 185 (DB2 data capture before and after image data). The

DB2 catalog must be accessed to obtain this information which describes data in log records.

**Tuning Tip:** The data capture facility can be used for hot site disaster recovery or sophisticated audit mechanisms. This field is associated with the time spent gathering the log record data in IFCID 185.

ELAPSED TIME-LOG EXTRACT

APPL CLASS 1
N/A

DB2 CLASS 2
N/A

IFI CLASS 5
Elapsed time extracting log data for IFCID 185 (QIFAAMLT).

This is the elapsed time processing log records as a result of IFI READS calls for IFCID 185 (data capture before and after image log data). This time is associated with the elapsed time processing data capture describes.

**Tuning Tip:** The data capture facility can be used for hot site disaster recovery or sophisticated audit mechanisms. This field is associated with the time spent gathering the log record data capture describes in IFCID 185.

DB2 EXIT/ENTRY EVENTS

APPL CLASS 1
N/A

DB2 CLASS 2
Number of exit/entry events (QWACARNA).

This value indicates the number of DB2 entry and exit events processed in determining the elapsed and processor times in DB2.

This field does not count entry and exit times for stored procedures which are gathered in a different counter. Accounting class 2 must be activated to determine this value.

IFI CLASS 5
N/A

CLASS 1 NOT NULL COUNT

APPL CLASS 1
Total number of accounting records with class 1 times and events.

DB2 CLASS 2
Total number of accounting records with class 2 times and events.

IFI CLASS 5
Total number of accounting records with class 5 times and events.

## QUERY PARALLELISM

PARALLEL DEGREE < PLANNED

Parallel group degraded due to buffer shortage (QXREDGRP).

This counter is incremented when the buffer pool does not have enough buffers to support as many degrees of parallel processing as had been planned. DB2 checks buffer allocations at both BIND and execution time. It assumes there will be buffers set aside for parallel processing. If at execution time a similar number of buffers do not exist, DB2 will degrade the parallel processes to a lesser degree or no parallelism.

**Tuning Tip:** Three parameters can be altered (ALTER BUFFERPOOL command) to resolve this situation. The overall size of the buffer pool is controlled by the VPSIZE (virtual pool size) parameter. The amount of sequential buffers is set by the VPSEQT (virtual pool sequential threshold) parameter. In the amount of buffers reserved for sequential processing, a reserve of buffers available for parallel processing must be maintained by the VPPSEQT (virtual pool parallel sequential threshold) parameter. If a significant number of parallel processes are degraded due to buffer shortage or contention, consider using a different buffer pool or altering the buffer pool used so that sufficient parallel sequential buffers are present.

PARALLEL GRPS FALLBACK-NO BUF

Parallel group fallback to sequential due to buffer shortage (QXDEGBUF).

This counter is incremented when the buffer pool does not have enough buffer storage in the virtual pool to support parallel processing. DB2 checks buffer allocations at both BIND and execution time. It assumes there will be buffers set aside for parallel processing. If at execution time a similar number of buffers do not exist, DB2 will degrade the parallel processes to a lesser degree or to no parallelism.

**Tuning Tip:** Three parameters can be altered (ALTER BUFFERPOOL command) to resolve this situation. The overall size of the buffer pool is controlled by the VPSIZE (virtual pool size) parameter. The amount of sequential buffers is set by the VPSEQT (virtual pool sequential threshold) parameter. In the amount of buffers reserved for sequential processing, a reserve of buffers available for parallel processing must be maintained by the VPPSEQT (virtual pool parallel sequential threshold) parameter. If a significant number of parallel processes are degraded due to buffer shortage or contention, consider using a different buffer pool or altering the buffer pool used so that sufficient parallel sequential buffers are present.

PARALLEL GRPS FALLBACK-CURSOR

Parallel group fallback to sequential due to updateable cursor (QXDEGCUR).

This counter is incremented when DB2 detects a cursor that is not clearly read-only and falls back from parallel processing to sequential.

**Tuning Tip:** To resolve this problem, the application program should have a cursor which is unambiguously read-only, with updates through another cursor or statement.

PARALLEL GRPS FALLBACK-NO ESA

Parallel group fallback to sequential due to lack of sort assist (QXDEGESA).

This counter is incremented when DB2 detects that the hardware sort assist facility is not present to logically partition the DB2 temporary DSNDB07 workfiles. The parallel sort operation falls back to sequential.

**Tuning Tip:** This situation will occur until the necessary hardware assist is purchased.

PARALLEL GRPS FALLBACK-ENCLAVE

Parallel group fallback to sequential due to unavailable enclave services (QXDEGENC).

This counter is incremented when DB2 detects that MVS 5.2 enclave support is unavailable to support parallel CP processing. The parallel group falls back to sequential.

**Tuning Tip:** MVS Enclave Support in MVS 5.2 sets objectives for parallel services to perform within service goals set by management in the MVS Workload Manager. The DB2 parallel tasks run as enclave SRBs. The solution is to migrate this system to MVS 5.2 as soon as practical to do so.

SINGLE DB2-COORDINATOR=NO

Number of parallel groups executed on a single DB2 due to one of the following reasons (QXCOORNO):

- When the plan or package was bound, the COORDINATOR subsystem parameter was set to YES, but the parameter is set to NO when the program runs.
- The plan or package was bound on a DB2 with the COORDINATOR subsystem parameter set to YES, but the program is being run on a different DB2 that has the COORDINATOR value set to NO.

(DB2 5.1 and later)

SINGLE DB2-ISOLATION=RR

Number of parallel groups executed on a single DB2 because the plan or package was bound with an isolation value of repeatable read or read stability (QXISORR) (DB2 5.1 and later).

DB2 MEMBER SKIPPED-NO BUF

Number of times the parallelism coordinator had to bypass a DB2 when distributing tasks because there was not enough buffer pool storage on one or more DB2 members (QXXCSKIP).

This field is incremented only on the parallelism coordinator, and it is incremented only once per parallel group, even though it is possible that more than one DB2 has a buffer pool shortage for that parallel group (DB2 5.1 and later).

**Tuning Tip:** The purpose of this count is to indicate that there are not enough buffers on one or more members. Therefore, this count is incremented only when the buffer pool is defined to allow parallelism. For example, if VPXPSEQT=0 on an assistant, DB2 does not send parallel work there, but this count is not incremented.

PARALLELISM DISABLED BY RLF

Number of times query parallelism was disabled by the Resource Limit Facility (RLF) for at least one dynamic select statement (QXRLFDPA).

PARALLEL GRPS PLANNED DEGREE

Number of parallel groups executed at planned degree (QXNORGRP).

This counter is incremented when the number of parallel tasks (degree) at execution time is the same as the number of parallel tasks planned at BIND time.

**Tuning Tip:** The higher this number is, the better DB2 is tuned. This is the ideal situation.

CPU PARALLEL TASKS CREATED

Number of parallel tasks created (QWACPCNT).

This is the number of parallel tasks an originating task created to process a query using query CP parallelism.

For a parallel task, the number will always be zero. Only the originating task increments this value. This value is calculated with accounting class 1.

PRLL GROUPS INTENDED

Number of parallel groups DB2 intended to run across the data sharing group (QXXCBPNX).

This count is incremented on the parallelism coordinator only at run time (DB2 5.1 and later).

PARALLEL GROUPS EXECUTED

Number of parallel groups executed (QXTOTGRP).

This counter reflects the total number of parallel groups executed.

**Tuning Tip:** This number can provide the tuner with a good idea of how often parallel processing was used for both read and sort activity.

MAX DEGREE OF PARALLELISM

Maximum degree of parallel processing executed (QXMAXDEG).

This counter is set to the high-water mark among all parallel groups executed for query parallelism.

The number reflects the plan that executed the highest degree of parallel processing among all parallel groups.

MAX MEMBERS PARTICIPATED

Represents the largest number of DB2 members that participated in processing a query across the sysplex (DB2 5.1 and later).

REFORM - CONFIG CHANGE

Total number of parallel groups for which DB2 reformulated the parallel portion of the access path because the sysplex configuration at run time was different from the sysplex configuration at bind time (QXREPOP1).

This counter is incremented only by the parallelism coordinator at run time.

(DB2 6.1 and later)

REFORM - NO BUFFER

Total number of parallel groups for which DB2 reformulated the parallel portion of the access path because there was not enough buffer pool resource (QXREPOP2).

This counter is incremented only by the parallelism coordinator at run time.

(DB2 6.1 and later)

## CLASS 3 SUSPENSION

LOCK/LATCH

Wait time for lock/latch (QWACAWTL).

This value indicates the elapsed time the thread waited for transaction locks and latches.

This time reflects the amount of time the thread was suspended because of an incompatible lock or latch mode. Accounting class 3 must be activated to determine this value.

SYNC I/O

Wait time for synchronous I/O (QWACAWTI).

This value indicates the elapsed time spent waiting for all synchronous I/O.

Most of the time included in this value is synchronous read time, but under stress conditions it can reflect synchronous write time (when immediate write threshold is reached). During heavy update activity, it also can reflect wait for synchronous log I/O. Accounting class 3 must be activated to determine this value.

**Note:** Log waits are not included in this value for DB2 6.1 and later.

OTHER THREAD READ I/O

Wait time for asynchronous reads (QWACAWTR).

This value is the time spent waiting for reads done under another thread to complete. These waits may be for sequential prefetch, sequential detection, list prefetch, or for synchronous I/O done while executing a sequential prefetch operation.

**Tuning Tip:** Time reflected here does not reflect the duration of the I/O but simply the time the thread waited for completion of the I/O activity. If this number is high, parallel processing may assist in reducing this time. Accounting class 3 must be activated to collect this number.

OTHER THREAD WRITE I/O

Wait time for write I/O (QWACAWTW).

This value is the time spent waiting for an asynchronous write I/O to complete or for synchronous write I/O to be performed for another thread.

**Tuning Tip:** Time reflected here should be small, since normal checkpoint and deferred write thresholds protect threads from waiting for asynchronous write time. If this number is large, the buffer pool size or deferred write threshold may need to be adjusted. Accounting class 3 must be activated to obtain this time.

EXECUTION TASK SWITCH

Wait time for DB2 services.

This value is the time spent waiting for special DB2 services. A synchronous execution unit switch is done from the thread to one of the DB2 service tasks.

Service times reported here are a summary of each of the following wait values:

- COMMIT/UPDATE (QWACAWTE)
- OPEN/CLOSE (QWAXOCSE)
- SYSLGRNG REC (QWAXSLSE)
- DEFINE/EXTEND/DELETE (QWAXDSSE)
- OTHER (QWAXOTSE)

Accounting class 3 must be activated to obtain these times.

EXECUTION TASK SWITCH - COMMIT/UPDATE

Accumulated wait time because of a synchronous execution unit switch for DB2 commit, abort, or deallocation processing (QWACAWTE) and the number of wait trace events processed for waits for synchronous execution unit switching for commit or abort (QWACARNS).

For DB2 5.1 and below, all the Execution Task Switch times are included in this value.

EXECUTION TASK SWITCH - OPEN/CLOSE

Accumulated wait time for a synchronous execution unit switch to the DB2 open/close data set service or the HSM recall service (QWAXOCSE) and the number of wait trace events processed for waits for synchronous execution unit switching to the open/close service (QWAXOCNS).

(DB2 6.1 and later)

EXECUTION TASK SWITCH - SYSLGRNG REC

Accumulated wait time for a synchronous execution unit switch to the DB2 SYSLGRNG recording service (QWAXSLSE) and the number of wait trace events processed for waits for synchronous execution unit switching to the SYSLGRNG recording service (QWAXSLNS).

This service is also sometimes used for level ID checking for down-level detection.

(DB2 6.1 and later)

EXECUTION TASK SWITCH - DEFINE/EXTEND/DELETE

Accumulated wait time for a synchronous execution unit switch to the DB2 data space manager services, which include define data set, extend data set, delete data set, reset data set, and VSAM catalog access (QWAXDSSE) and the number of wait trace events processed for waits for synchronous execution unit switching to the data space manager service tasks (QWAXDSNS).

(DB2 6.1 and later)

EXECUTION TASK SWITCH - OTHER

Accumulated wait time for a synchronous execution unit switch to other DB2 service tasks (QWAXOTSE) and the number of wait trace events processed for waits for synchronous execution unit switching to other service tasks (QWAXOTNS).

(DB2 6.1 and later)

ARCHIVE LOG QUIESCE

Wait time for log quiesce (QWACALOG or QWAXALOG in DB2 6.1 and later).

This value is the time spent waiting for an -ARCHIVE LOG MODE(QUIESCE) to take place.

This process externalizes buffers and switches the active log data set. Time reflected here is the time the thread waited not the time for the command to complete. Accounting class 3 must be activated to obtain this time.

ARCHIVE LOG READ

Wait time for archive read from tape (QWACAWAR or QWAXAWAR in DB2 6.1 and later).

This is the amount of time spent waiting for reads from an archive tape.

Time spent waiting for an archive tape is either for the completion of the RECOVER utility or for an application program, which has not committed, to read the archive tape in order to complete backout. Accounting class 3 must be activated to obtain this time.

**Tuning Tip:** If this time is large, the application program should commit more often. It may also make sense to increase the size and number of active logs.

DRAIN LOCK

Wait time for drain lock (QWACAWDR or QWAXAWDR in DB2 6.1 and later).

This value is the time spent by a requester of a drain lock, such as a utility or command, waiting to acquire that lock.

Processes, such as utilities that serialize events to page sets, must acquire a drain lock. This is the time spent waiting for this event. Accounting class 3 must be activated to obtain this time.

CLAIM RELEASE

Wait time for claim release (QWACAWCL or QWAXAWCL in DB2 6.1 and later).

This value is the time spent by the holder of a drain lock waiting for SQL claimers to complete their activity.

Drainers must wait until the SQL claim count reaches zero. This is the time spent waiting for them to clear. Accounting class 3 must be activated to obtain this time.

PAGE LATCH CONTENTION

Wait time for page latch contention (QWACAWTP).

This is the time spent waiting for page latching activities due to lock avoidance. Accounting class 3 must be activated to obtain this time.

**Tuning Tip:** Page latching can be avoided by scheduling applications when there is a lower likelihood of latch contention.

STORED PROCEDURE

Wait time for SPAS TCB scheduling (QWACCAST).

This time is the elapsed time spent waiting for an available TCB in the Stored Procedures Address Space to become available for scheduling a stored procedure.

**Tuning Tip:** This time should be minimal depending on the workload to the Stored Procedures Address space. It is calculated only if accounting class 3 is active.

NOTIFY MESSAGES

Wait time for data sharing notify messages (QWACAWTG).

This value is the amount of wait time for sending notify messages to another data sharing member.

When tables in a data sharing group are CREATEd, ALTERed, or DROPped in a data sharing group, intersystem messages must be sent to update the database descriptors (DBDs) in the other members’ EDM pools. This is the time spent waiting for this message traffic to occur. This time is collected only when accounting class 3 is activated and data sharing is enabled.

GLOBAL CONTENTION

Wait time for global lock contention (QWACAWTJ).

This time is the amount of elapsed time caused by an IRLM lock suspension due to global lock contention which requires intersystem communication to resolve.

This time is collected only when accounting class 3 is activated and data sharing is enabled.

CLASS 3 TOTAL WAIT

Total elapsed time waiting for all reported events listed above.

CLASS 3 NOT NULL COUNT

Total number of accounting records with class 3 times or events.

## LOCKING ACTIVITY

TIMEOUTS

Number of lock timeouts (QTXATIM).

This number is incremented every time a DB2 thread waits to get a resource longer than the timeout interval. It is specified with the DSNZPARM IRLMRWT on installation panel DSNTIPI. By default, it is 60 seconds. Utilities can be allowed to wait several multiples of IRLMRWT.

**Tuning Tip:** Lock timeouts are usually caused by an application failing to commit in time for the other thread to gain access to data on pages. Often the problem can be resolved by reducing the time between commits and putting updateable statements near to their COMMIT logic. Every time a timeout occurs, DB2 writes the holder and suspender to the MSTR job log. Normally, this number should be as close to zero as possible.

DEADLOCKS

Number of deadlocks (QTXADEA).

This counter is incremented every time DB2 encounters a deadlock situation for which the IRLM must cancel a task involved in a *deadly embrace*.

**Tuning Tip:** Deadlocks are caused by threads requesting access to two resources which can never be resolved. DB2 chooses its victim by selecting the thread which has done the least number of updates. It records the deadlock in the MSTR address job log. In well-tuned systems, this number should be low. The most frequent cause of deadlock situations are ascending key indexes. The Type 2 indexes available in DB2 4.1 can resolve many of these problems because no index locks are taken.

LOCK SUSPENDS

Number of suspends due to lock conflict (QTXASLOC).

This counter is incremented any time a thread has a conflicting lock request, such as an updater requesting exclusive access to a page another thread is using.

**Tuning Tip:** In a multitasking system, suspensions occur in the normal course of the events. If applications are well-tuned, taking frequent commits and holding on to resources for the fewest possible instructions can minimize suspensions. If suspensions cause frequent timeouts, consider row-level locking.

OTHER SUSPENDS

Number of suspends due to other conflicts (QTXASOTH).

This number is incremented when DB2 internal processes collide.

**Tuning Tip:** This number is not generally of significance in tuning.

LATCH SUSPENDS

Number of suspends due to latch conflicts (QTXASLAT).

This number is incremented when a latch conflict exists between two DB2 threads or internal serialization processing takes place.

**Tuning Tip:** Latches are generally of extremely short duration. Unless the time is a significant component of overall wait time, it is not a factor which should cause tuning problems.

MAXIMUM PAGE LOCKS

Maximum number of page or row locks held (QTXANPL).

This counter represents the highest number of page or row locks held during an execution of a plan.

**Tuning Tip:** This number may not exceed the NUMLKUS (number of locks per user) count in DSNZPARM. If it does, the user will get a -904 resource unavailable message with a 00C90096 reason code. This number is a significant tuning knob which should be observed when migrating an application from one system to another, particularly if the target DB2 has a different number for the maximum locks threshold. The application can free locks by committing resources more often. If LOCKSIZE(ROW) is specified, DB2 holds a lock for each row on the page. Depending on the row size, this can be extremely costly. Consider the cost of row-level locking carefully before implementing.

LOCK REQUESTS

Lock request count (QTXALOCK).

This counter is incremented for each call to the IRLM lock manager to acquire a lock on a page or row or to acquire a claim or drain on a data set.

**Tuning Tip:** Each lock request is processed by the IRLM. Lock avoidance techniques should show reductions in overall counts and overhead, since latches execute totally within DB2.

LOCK QUERY REQUESTS

Query request count (QTXAQRY).

This counter is incremented every time the IRLM gets a request to read data.

**Tuning Tip:** This information is useful in determining the read activity but does not include lock avoidance techniques.

LOCK CHANGE REQUESTS

Change request count (QTXACHG).

This counter is incremented every time the IRLM is asked to change a lock from one type to another (for example, from S to X).

OTHER IRLM REQUESTS

Other IRLM request count (QTXAIRLM).

This counter is incremented every time the IRLM receives a lock request not included in the other counts.

UNLOCK REQUESTS

Unlock request count (QTXAUNLK).

This counter is incremented when the application has finished processing the page or row or when a claim or drain can be released.

This amount is significant as to cross memory processing and reflects the normal release of resources.

LOCK ESCALATIONS TO SHARED

Number of lock escalations to shared mode (QTXALES).

This counter is incremented every time the number of locks against a single table space exceeds the number set in DSNZPARM NUMLKTS on installation panel DSNTIPJ or the number set in the LOCKMAX clause of the CREATE TABLESPACE statement.

**Tuning Tip:** This is not a normal situation unless using repeatable read. If it occurs often, consider changing the LOCKSIZE or LOCKMAX to a higher value, or consider binding the plan with cursor stability or uncommitted read (UR).

LOCK ESCALATIONS TO EXCLUSIVE

Number of lock escalations to exclusive mode (QTXALEX).

This counter is incremented every time the number of updateable locks against a single table space exceeds the DSNZPARM NUMLKTS on installation panel DSNTIPJ or in the LOCKMAX clause of the CREATE TABLESPACE statement. It occurs when the LOCKSIZE parameter is specified as ANY and DB2 has escalated the lock owner to an exclusive lock of the entire table.

**Tuning Tip:** This is an extremely undesirable situation, usually caused by leaving the LOCKSIZE(ANY) default. To resolve this situation, consider changing the parameter to LOCKSIZE (PAGE) or, in special situations, LOCKSIZE(ROW). This will cause the offending application to take the -904 unavailable resource error rather than cause general unavailability to the rest of the users. This is almost always caused by application failure to commit in a timely fashion and can be resolved by application code changes as well as by DBA action.

LOCK TOTALS

Total number of lock events listed above.

## DRAIN / CLAIM

CLAIM REQUESTS

Number of claim requests (QTXACLNO).

This number is incremented every time a user executes an SQL statement that increments the use count of a table space, partition, or index space.

FAILED CLAIM REQUESTS

Number of unsuccessful claim requests (QTXACLUN).

This number is incremented every time a user issues a request for a claim to an SQL resource but cannot acquire one, usually because a utility or command DRAIN is on the object being sought.

**Tuning Tip:** This number is of some significance in determining contention between SQL and other types of utilities or commands.

DRAIN REQUESTS

Number of drain requests (QTXADRNO).

This counter is incremented each time a utility or command requests a serialization against a page set resource.

**Tuning Tip:** This is of significance in determining the amount of utility and command activity which request some serial drain access to a resource.

FAILED DRAIN REQUESTS

Number of unsuccessful drain requests (QTXADRUN).

This counter is incremented when a potential drainer (utility or command) cannot obtain use of a page set because the claim count has not dropped to zero within the utility timeout value set in IRLMWRT of DSNZPARMs.

**Tuning Tip:** This number is of significance in determining the number of unsuccessful utility and command processes due to user activity on the resource.

## OPTIMIZATION

These fields apply only to DB2 5.1 and later.

REOPTIMIZATION
Number of times reoptimization occurred for a query (QXSTREOP).

PREPARE - SATISFIED
Number of times DB2 satisfied a PREPARE request by making a copy of a statement in the prepared statement cache (QXSTFND).

PREPARE - FAILED SEARCH
Number of times DB2 searched the prepared statement cache but could not find a suitable prepared statement (QXSTNFND).

PREPARE - IMPLICIT
Number of times DB2 did an implicit PREPARE for a statement bound with KEEPDYNAMIC(YES) because the prepared statement cache did not contain a valid copy of the prepared statement (QXSTIPRP).

PREPARE - FROM CACHE
Number of times DB2 did not prepare a statement bound with KEEPDYNAMIC(YES) because the prepared statement cache contained a valid copy of the prepared statement (QXSTNPRP).

PREPARE - DISCARDED MAX
Number of times DB2 discarded a prepared statement from the prepared statement cache because the number of prepared statements in the cache exceeded the value of subsystem parameter MAXKEEPD (QXSTDEXP).

PREPARE - DROP/ALTER/REVOKE
Number of times DB2 discarded a prepared statement from the prepared statement cache because a program executed a DROP, ALTER, or REVOKE statement against a dependent object (QXSTDINV).

## RID LIST

RID LIST PROCESSING USED

Number of times RID pool used (QXMIAP).

This counter is incremented every time the RID pool in the database services address space is used. The RID pool is always used to sort index keys by the RBA so that list prefetch (read-ahead buffering) can be enabled against data pages.

**Tuning Tip:** RID pool usage and sizing is critical for effective DB2 performance. The RID pool size is set by the systems installer in DSNZPARM SPRMRDP on the DSNTIPC installation panel.

The DB2 RID pool is used for two purposes:

- To use multiple indexes to access a table

  In this case, the index entries are read, sorted into RID sequence, unioned (for OR processing), or intersected (for AND processing). Then the data from the table is accessed using list prefetch.

- To resolve an answer set from a non-matching index scan or a nested loop join

  Again, the RIDs are sorted into RID sequence and the data is read using list prefetch.

Failure of RID pool storage causes DB2 to revert to table space scan processing. For large tables, this can cause extremely poor performance.

DB2 will also end list prefetching as inefficient if more than 25% of the rows in the table must be accessed.

RID FAILURES - OVER MAX

Number of RID pool failures due to exceeding internal limits (QXMRMIAP).

This counter is incremented when DB2 detects that a single index would exceed 50% availability in the RID pool for a multiple index access or list prefetch request. The field can be incremented at any stage (retrieval, sorting, ANDing/ORing of RID lists for non-matching index scan processing) or incremented when each index of a multiple index access scan finds no more storage is available.

**Tuning Tip:** DB2 attempts to protect the RID pool from a large result set that would exceed 50% of the pool or reach the maximum size of 16 million RIDs for a single RID; in other words, it will cause a RID pool failure if the number of RIDS from any single index or any index involved in a multiple index access attempt exceeds 50%. If this happens often, attempt to find the offending application and change the SQL or increase the size of the RID pool.

RID FAILURES - STORAGE

Number of RID pool failures due to no storage (QXNSMIAP).

This counter is incremented when DB2 detects that no storage is available in the RID pool for a multiple index access or list prefetch request. The field can be incremented at any stage (retrieval, sorting, ANDing/ORing of RID lists for non-matching index scan processing) or incremented when each index of a multiple index access scan finds no more storage is available.

**Tuning Tip:** This is an extremely undesirable situation as the portion of the query which would use the index falls back to table scan processing. If this occurs often, the size of the RID pool specified in SPRMRDP of the DSNZPARMs (on installation panel DSNTIPC) should be increased.

## Accounting Long Report—Page 2

The fields in this section are organized in the following categories. For the highlights, see page 130.

### SQL DML

SELECT
Number of embedded SELECT statements (QXSELECT).

This is a count of the number of single embedded SELECT statements processed for the application.

INSERT
Number of INSERT statements (QXINSRT).

This counter is incremented when an INSERT statement is processed for the application.

UPDATE
Number of UPDATE statements (QXUPDTE).

This counter is incremented when an UPDATE statement is processed for the application.

DELETE
Number of DELETE statements (QXDELET).

This counter is incremented when a DELETE statement is processed for the application.

PREPARE
Number of PREPARE statements (QXPREP).

This counter is incremented when the SQL PREPARE statement (which mini-binds a dynamic SQL statement) is processed for the application.

The count between a server and requester may not be equal.

DESCRIBE
Number of DESCRIBE statements (QXDESC).

This counter is incremented when a DESCRIBE statement (for dynamic or distributed processing) is processed for the application.

The count between a server and requester may not be equal.

DESCRIBE TABLE
Number of DESCRIBE TABLE statements (QXDSCRTB).

This counter is incremented every time a DESCRIBE TABLE statement is executed.

OPEN
Number of OPEN CURSOR statements (QXOPEN).

This counter is incremented when the OPEN CURSOR statement is processed for the application.

FETCH
Number of FETCH statements (QXFETCH).

This counter is incremented every time a FETCH statement is executed from a cursor.

CLOSE

Number of CLOSE CURSOR statements (QXCLOSE).

The counter is incremented when a CLOSE CURSOR statement is processed for the application.

The count between a server and requester may not be equal.

DML TOTALS

Total number of SQL DML statements processed.

## SQL DCL

LOCK TABLE
Number of LOCK TABLE statements (QXLOCK).

This counter is incremented every time a LOCK TABLE statement is executed (either SHARE or EXCLUSIVE).

GRANT
Number of GRANT statements (QXGRANT).

This counter is incremented every time a GRANT statement is executed.

**Tuning Tip:** This field is useful for the auditor who wishes to monitor the grants of authority by user. Other audit traces can be activated to capture which authorities were granted. The catalog can also be queried.

REVOKE
Number of REVOKE statements (QXREVOK).

This counter is incremented every time a REVOKE statement is executed.

**Tuning Tip:** This field is useful for the auditor who wishes to monitor the revocations of authority by user. Other audit traces can be activated to capture which authorities were revoked.

SET CURRENT SQLID
Number of SET CURRENT SQLID statements (QXSETSQL).

This counter is incremented every time a SET CURRENT SQLID statement is executed.

SET HOST VARIABLES
Number of SET HOST VARIABLE statements (QXSETHV).

This counter is incremented every time a SET HOST VARIABLE statement is executed.

SET CURRENT DEGREE
Number of SET CURRENT DEGREE statements executed (QXSETCDG).

This is a count of the number of SQL SET CURRENT DEGREE statements processed for the application. This register enables or disables parallel processing for dynamic SQL.

SET CURRENT RULES
Number of SET CURRENT RULES statements executed (QXSETCRL).

This is a count of the number of SQL SET CURRENT RULES statements processed for the application. This register is used to change syntax parsing from SQL rules to ANSI/SQL processing.

SET CURRENT PATH
Number of SET CURRENT PATH statements executed (QXSETPTH).

(DB2 6.1 and later)

CONNECT TYPE 1

Number of Type 1 CONNECT statements executed (QXCON1).

This is a count of the number of Type 1 SQL CONNECT statements processed for the application.

**Tuning Tip:** Type 1 SQL CONNECT statements allow one updateable site in the connection. Type 2 SQL CONNECT statements allow multiple updateable sites in the distributed connection. The type of CONNECT statement is specified as a precompiler parameter CONNECT(1) or CONNECT(2).

CONNECT TYPE 2

Number of Type 2 CONNECT statements executed (QXCON2).

This is a count of the number of Type 2 CONNECT SQL statements processed for the application.

**Tuning Tip:** Type 1 SQL CONNECT statements allow one updateable site in the connection. Type 2 SQL CONNECT statements allow multiple updateable sites in the distributed connection. The type of CONNECT statement is specified as a precompiler parameter CONNECT(1) or CONNECT(2).

SET CONNECTION

Number of SET CONNECTION statements executed (QXSETCON).

This is a count of the number of SQL SET CONNECTION statements processed for the application.

**Tuning Tip:** This verb is required to perform multi-site updates.

RELEASE

Number of RELEASE statements issued (QXREL).

This is a count of the number of SQL RELEASE statements processed for the application.

CALL

Number of SQL CALL statements executed (QXCALL).

This is a count of the number of SQL CALL statements processed for the application. This activates a DB2 stored procedure.

ASSOCIATE LOCATOR

Number of ASSOCIATE LOCATOR statements executed (QXALOCL).

These statements get the result set locator value for each result set returned by a stored procedure (DB2 5.1 and later).

HOLD LOCATOR

Number of HOLD LOCATOR statements executed (QXHOLDL).

(DB2 6.1 and later)

FREE LOCATOR

Number of FREE LOCATOR statements executed (QXFREEL).

(DB2 6.1 and later)

ALLOCATE CURSOR

Number of ALLOCATE CURSOR statements executed (QXALOCC).

These statements define a cursor and associate it with the result set locator variable (DB2 5.1 and later).

DCL TOTALS

Total number of SQL DCL statements processed.

## SQL DDL CREATE

CREATE STORAGE GROUP
Number of CREATE STOGROUP statements (QXCRSTG).

This counter is incremented every time a CREATE STORAGE GROUP statement is executed.

CREATE DATABASE
Number of CREATE DATABASE statements (QXCRDAB).

This counter is incremented every time a CREATE DATABASE statement is executed.

CREATE TABLESPACE
Number of CREATE TABLESPACE statements (QXCTABS).

This counter is incremented every time a CREATE TABLESPACE is executed.

CREATE TABLE
Number of CREATE TABLE statements (QXCRTAB).

This counter is incremented every time a CREATE TABLE statement is executed.

A QMF SAVE DATA performs a CREATE TABLE. The tuner can monitor how many tables have been created by QMF users.

CREATE INDEX
Number of CREATE INDEX statements (QXCRINX).

This counter is incremented every time a CREATE INDEX statement is executed.

CREATE VIEW
Number of CREATE VIEW statements (QXDEFVU).

This counter is incremented every time a CREATE VIEW statement is executed.

CREATE SYNONYM
Number of CREATE SYNONYM statements (QXCRSYN).

This counter is incremented every time a CREATE SYNONYM statement is executed.

CREATE ALIAS
Number of CREATE ALIAS statements (QXCRALS).

This counter is incremented every time a CREATE ALIAS statement is executed.

CREATE TEMP TABLE
Number of CREATE TEMPORARY TABLE statements (QXCRGTT).

This counter is incremented every time a CREATE TEMPORARY TABLE statement is executed (DB2 5.1 and later).

CREATE AUX TABLE
Number of CREATE AUXILIARY TABLE statements (QXCRATB).

(DB2 6.1 and later)

CREATE PROCEDURE
Number of CREATE PROCEDURE statements (QXCRPRO).

(DB2 6.1 and later)

CREATE FUNCTION
Number of CREATE FUNCTION statements (QXCRUDF).

(DB2 6.1 and later)

CREATE TRIGGER
Number of CREATE TRIGGER statements (QXCTRIG).

(DB2 6.1 and later)

CREATE DISTINCT TYPE
Number of CREATE DISTINCT TYPE statements (QXCDIST).

(DB2 6.1 and later)

DDL CREATE TOTALS
Total number of SQL DDL CREATE statements processed.

## SQL DDL ALTER

ALTER STORAGE GROUP
Number of ALTER STOGROUP statements (QXALTST).

This counter is incremented every time an ALTER STOGROUP statement is executed.

ALTER DATABASE
Number of SET ALTER DATABASE statements (QXALDAB).

This counter is incremented every time an ALTER DATABASE statement is executed.

ALTER TABLESPACE
Number of ALTER TABLESPACE statements (QXALTTS).

This counter is incremented every time an ALTER TABLESPACE statement is executed.

ALTER TABLE
Number of ALTER TABLE statements (QXALTTA).

This counter is incremented every time an ALTER TABLE statement is executed.

ALTER INDEX
Number of ALTER INDEX statements (QXALTIX).

This counter is incremented every time an ALTER INDEX statement is executed.

ALTER PROCEDURE
Number of ALTER PROCEDURE statements (QXALPRO).

(DB2 6.1 and later)

ALTER FUNCTION
Number of ALTER FUNCTION statements (QXALUDF).

(DB2 6.1 and later)

DDL ALTER TOTALS
Total number of SQL DDL ALTER statements processed.

RENAME TABLE
Number of RENAME TABLE statements (QXRNTAB).

This counter is incremented every time a RENAME TABLE statement is executed (DB2 5.1 and later).

COMMENT ON
Number of COMMENT ON statements (QXCMTON).

This counter is incremented every time a COMMENT ON statement is executed.

**Tuning Tip:** This field is useful for determining whether documentation was done during a particular time period (COMMENT ON simply puts comments into the DB2 catalog).

LABEL ON
Number of LABEL ON statements (QXLABON).

This counter is incremented every time a LABEL ON statement is executed.

## SQL DDL DROP

DROP STORAGE GROUP
Number of DROP STOGROUP statements (QXDRPST).

This counter is incremented every time a DROP STOGROUP statement is executed.

DROP DATABASE
Number of DROP DATABASE statements (QXDRPDB).

This counter is incremented every time a DROP DATABASE statement is executed.

DROP TABLESPACE
Number of DROP TABLESPACE statements (QXDRPTS).

This counter is incremented every time a DROP TABLESPACE statement is executed.

DROP TABLE
Number of DROP TABLE statements (QXDRPTA).

This counter is incremented every time a DROP TABLE statement is executed.

DROP INDEX
Number of DROP INDEX statements (QXDRPIX).

This counter is incremented every time a DROP INDEX statement is executed.

DROP VIEW
Number of DROP VIEW statements (QXDRPVU).

This counter is incremented every time a DROP VIEW statement is executed.

DROP PACKAGE
Number of DROP PACKAGE statements (QXDRPPKG).

This counter is incremented every time a DROP PACKAGE statement is executed.

DROP SYNONYM
Number of DROP SYNONYM statements (QXDRPSY).

This counter is incremented every time a DROP SYNONYM statement is executed.

DROP ALIAS
Number of DROP ALIAS statements (QXDRPAL).

This counter is incremented every time a DROP ALIAS statement is executed.

DROP PROCEDURE
Number of DROP PROCEDURE statements (QXDRPPR).

(DB2 6.1 and later)

DROP FUNCTION
Number of DROP UDF statements (QXDRPFN).

(DB2 6.1 and later)

DROP TRIGGER

Number of DROP TRIGGER statements (QXDRPTR).

(DB2 6.1 and later)

DROP DISTINCT TYPE

Number of DROP DISTINCT TYPE statements (QXDDIST).

(DB2 6.1 and later)

DDL DROP TOTALS

Total number of SQL DDL DROP statements processed.

## DATA CAPTURE

DATA CAP DESCRIPTION RETURNED

Number of data capture descriptions returned (QIFAANDD).

This reflects the number of catalog entries described in IFCID 185 to allow parsing of the row data. The DB2 catalog must be accessed to obtain this information. The fields are mapped in QW0185DD.

DATA ROW RETURNED - IFCID185

Number of data rows returned in IFCID 185 (QIFAANDR).

Before and after images of each row are returned to the user. The rows are mapped by macro QW0185DR.

IFI ENTRIES - CLASS 5

Number of entries and exits from IFI events (QIFAANIF).

This value is the number of times DB2 processed trace data records from the IFI. This information is obtained only if accounting class 5 is active.

DATA CAPTURE LOG READS

Number of log reads performed for data capture (QIFAANLR).

This counter is incremented every time the DB2 log has to be read for an IFI READS request to process data capture changes in IFCID 185.

Data Capture logs the entire row of before and after change data. This number represents how many times the log had to be read to provide this data for processing IFI READS requests.

DATA CAP DESCRIBES

Number of describes done for data capture (QIFAANMB).

This counter is incremented every time DB2 must go to the catalog to describe a row for processing READS requests for IFCID 185.

This facility is available for the data capture exit which is used for fast remote site disaster recovery or sophisticated audit techniques.

DATA CAPTURE LOG REC

Number of log records written for data capture (QIFAANRC).

This counter is incremented when log records are written to IFCID 185. It includes only those log records that can be retrieved with an IFI READS call for IFCID 185 (Data Capture).

**Tuning Tip:** These records are gathered by ALTERing a table space to CAPTURE YES.

LOG REC RETURNED - IFCID185

Number of data capture records returned to caller (QIFAANRR).

The counter is incremented when IFCID 185 is successfully returned to the caller.

**Tuning Tip:** Data capture records are used for remote hot site recovery and sophisticated audit software.

DATA CAP TABLES RETURNED

Number of tables returned to data capture caller (QIFAANTB).

This counter is incremented for each table described in a data capture IFCID 185 trace record. This requires a catalog lookup. The data is used in decoding changed information in the data capture exit.

## SERVICE UNITS

This section is available for DB2 5.1 and later only.

The following service units (SUs) have been converted from CPU times using the conversion factor from QWACSUCV as follows:

SU = CPU seconds x (16000000/conversion factor)

SERVICE UNITS - TOTAL
Total CPU time converted to service units.

SERVICE UNITS - TCB
TCB CPU time converted to service units.

SERVICE UNITS - STORED PROCEDURE
CPU time spent processing SQL stored procedure CALL statements converted to service units.

SERVICE UNITS - TRIGGER
CPU time spent processing triggers converted to service units.

(DB2 6.1 and later)

SERVICE UNITS - USER FUNCTION
CPU time spent processing user-defined functions converted to service units.

(DB2 6.1 and later)

SERVICE UNITS - PARALLEL TASKS
CPU time spent processing parallel tasks converted to service units.

## Accounting Long Report—Page 3

The fields in this section are organized in the following categories. For the highlights, see page 130.

### BUFFER POOL

BPOOL HIT RATIO(%)

Buffer pool hit ratio with prefetch.

This value is calculated as

(Total Pages - Total I/O) / Total Pages x 100.0

where

Total Pages is the number of getpage requests (QBACGET).

Total I/O is the sum of

- Synchronous read I/Os (QBACRIO)
- Asynchronous pages read by sequential prefetch (QBACSIO)

GETPAGES

Number of getpage requests (QBACGET).

This is a count of the number of requests for a data page (successful and unsuccessful) from DB2's data manager to the buffer manager for normal DB2 processing.

DB2 looks first in a central storage virtual buffer pool, then to an expanded storage hiperpool, if present. If not present, DB2 reads the page from DASD. In parallel query processing, the field counts only the number of successful requests.

**Tuning Tip:** The general rule of thumb for tuning any query is to attempt to find the page in memory if possible, either by maintaining large pools to improve the hit ratio for random reads or by use of sequential prefetch, dynamic detection, or list prefetch to improve sequential reads. The higher the ratio of pages found in memory, the less I/O the query costs and the faster the query will run.

SYNCHRONOUS READ I/O

Number of synchronous read I/Os (QBACRIO).

This counter is incremented every time DB2 cannot find a page in memory and must issue a random I/O to DASD to retrieve it.

**Tuning Tip:** The overall goal in DB2 tuning is to reduce the number of synchronous I/Os by having needed pages in memory, either by caching or use of prefetch. The thread waits for synchronous I/O activity to be completed before proceeding. This wait time is measured with accounting class 3 and is reflected in the SYNC I/O field (see page 139 (thread) or page 177 (package)).

GETPAGES / READIO

Number of getpages per synchronous read I/O (QBACGET / QBACRIO).

This is the buffer hit ratio for random access applications (prefetch pages read are not included).

SYNCHRONOUS WRITE I/O

Number of immediate write I/Os (QBACIMW).

This counter is the number of synchronous write I/Os.

This situation only occurs when the immediate write threshold of 97.5% of pages in use in a buffer pool is reached. This situation is extremely resource intensive and highly undesirable. To avoid reaching this threshold, the VPSIZE (virtual pool size) should be tuned so this never occurs.

ASYNC PAGES READ

Number of asynchronous pages read (QBACSIO).

This count is the number of pages read by prefetch operations.

**Tuning Tip:** Generally, unless the application is totally random, the higher the prefetch number, the more likely it is that the data will be in memory when needed. The more successful read-ahead buffering is, the faster the application should perform.

SEQUENTIAL PREFETCH

Number of sequential prefetch requests (QBACSEQ).

This counter is incremented each time DB2 issues a sequential prefetch request. Normally DB2 attempts to do this read-ahead buffering by reading up to 32 pages per prefetch request and up to 64 pages per utility prefetch.

**Tuning Tip:** A buffer pool must be at least 1000 pages to get the full benefit of sequential prefetch. Also, a sufficient number of sequential pages must be allowed in the buffer pool assigned to the accessed table (VPSEQT parameter). Otherwise, the prefetch quantity may be reduced, or prefetch disabled.

LIST PREFETCH

Number of list prefetch requests (QBACLPF).

This counter is incremented each time DB2 reads index RIDs (Row IDs) in non-matching index scans, multiple index access path selections, or several types of join access paths.

List prefetch allows DB2 to sort the index RIDs into data row order, thereby enabling prefetch in data sequence, a process which eliminates data page re-reads and allows read-ahead buffering. The RIDs are sorted in the RID pool.

**Tuning Tip:** List prefetch is not chosen at BIND if DB2 determines that the resultant size of the RID list would exceed 50% of the RID pool. At execution time, DB2 disables list prefetch if a single index occupies more than 25% of the RID pool or the RID pool storage is exhausted. When list prefetch is disabled, the query becomes a table space scan. This is highly undesirable. If it occurs frequently, change the SQL to reduce the number of RIDs or increase the size of the RID pool.

DYNAMIC PREFETCH

Number of dynamic prefetch requests (QBACDPF).

Dynamic prefetch is triggered by DB2 sequential detection. If DB2 determines that a cursor is moving sequentially through the data, it automatically enables sequential prefetch.

This process allows DB2 to improve performance when sequential prefetch was not enabled at BIND but the data is being retrieved sequentially. DB2 examines the data being retrieved and if it determines that the last five of eight pages are sequential, it uses prefetch to request the next set of up to 32 pages until DB2 determines the criteria are no longer met.

PAGES UPDATED

Number of page updates (QBACSWS).

This counter is incremented every time a page is updated and ready to be externalized to DASD. If the same page is updated twice, for example, the count is incremented by 2. This count includes not only updates to data pages but also workfile pages, so if a sort is used, this number can be higher than expected.

DB2 keeps committed pages in a buffer pool until a system checkpoint occurs or the deferred write threshold is encountered. DB2 attempts to write data to the page set asynchronously.

The number of pages used in prefetch is governed by the size of the buffer pool. Prefetch can be 8, 16, or 32 pages.

HIPERPOOL ASYNC READS

Number of asynchronous hiperpool pages read (QBACHPG).

This counter reflects the number of pages found in a hiperpool and moved to a virtual buffer pool because of a prefetch operation.

Pages can be moved from hiperpool to virtual pools very quickly; if, as a result of sequential prefetch, the hardware Asynchronous Data Mover Facility (ADMF) is used, the process is very efficient.

HIPERPOOL SYNC READS

Number of synchronous moves from hiperpool to virtual pool (QBACHRE).

This counter is incremented each time DB2 needs a page in a virtual (central) buffer pool and determines that it is stored in an expanded storage hiperpool.

**Tuning Tip:** DB2 issues a MVPG (movepage) instruction to move the data from expanded to central storage. Updates can occur only in central storage. The process is synchronous in that the thread waits for the activity to complete, but the hiperpool access times are in microseconds (millionths of seconds) compared to a DASD read (thousandths of seconds).

HIPERPOOL READ FAIL

Number of synchronous hiperpool reads that fail due to MVS page stealing (QBACHRF).

This number is incremented when DB2 finds a page in the hiperpool but MVS has stolen the buffer.

**Tuning Tip:** This occurs only if the CASTOUT parameter for that buffer pool is set to YES. DB2 must then reread the page from DASD. If the number is consistently high, it indicates the hiperpool has serious competition for MVS resources.

It would be appropriate to reduce the size of the hiperpool unless the application is so critical that it is deemed necessary to change CASTOUT to NO. CASTOUT NO essentially takes pages from expanded storage and makes them unavailable to MVS. This is generally not desirable as it may seriously degrade MVS performance for other workloads.

HIPERPOOL WRITES

Number of pages written synchronously to the hiperpool (QBACHWR).

This number is incremented each time DB2 moves a page from a central storage buffer pool to expanded storage. This indicates successful caching of a a page.

If a hiperpool were not available, the page would be discarded. By moving the page contents to cheaper expanded storage, DB2 is increasing the effectiveness of data page reuse. Before reusing an updated page DB2 must move the new contents to expanded storage since updates never take place in a hiperpool.

HIPERPOOL WRITES FAIL

Number of pages not written to hiperpool because of expanded storage shortage (QBACHWF).

This number is incremented when a shortage of hiperpool space exists and DB2 cannot move the pages from central to expanded storage. This occurs when serious MVS demands are stealing the hiperpool pages. When this occurs, pages are aged out from the central virtual buffer pool as they would have without a hiperpool.

**Tuning Tip:** Consider reducing the size of the hiperpool so that all hiperpools can co-exist with MVS requirements.

## GLOBAL BUFFER POOL

This section is available for DB2 4.1 and later only.

SYNC RD INV BUFF

Number of synchronous coupling facility reads for *invalid buffer* in each of these circumstances:

- WITH DATA

A data sharing coupling facility synchronous read was issued because a local virtual buffer pool or hiperpool had a page marked as invalid but the page existed in the group buffer pool (QBGAXD).

This occurs normally in a data sharing complex in which data must be shared. The correct data is refreshed from the group buffer pool.

- NO DATA

A data sharing coupling facility synchronous read was issued because a local virtual buffer pool or hiperpool had a page marked as invalid but no data was returned (QBGAXR + QBGAXN).

SYNC RD NOT FOUND

Number of synchronous coupling facility reads for *buffer not found* in each of these circumstances:

- WITH DATA

A data sharing coupling facility synchronous read was issued because a page was not found in the local virtual buffer pool or hiperpool but the page existed in the group buffer pool (QBGAMD).

This occurs normally in a data sharing complex in which data must be shared. The correct data is refreshed from the group buffer pool.

- NO DATA

A data sharing coupling facility synchronous read was issued because a page could not be found in a local virtual buffer pool or hiperpool and no data was returned (QBGAMR + QBGAMN).

PAGES WRITTEN

Number of pages written to the group buffer pool in each of these circumstances:

- CHANGED

A changed page was written to the group buffer pool (QBGASW).

Updated pages must be written to the coupling facility when the object is of intersystem interest so other DB2s can refresh their invalidated buffers. Only changed pages are written when GBPCACHE is set to CHANGE.

- CLEAN

A clean page was written to the group buffer pool (QBGAWC).

This is done only when GBPCACHE is set to ALL.

**Tuning Tip:** This can be expensive but does post the group buffer pool with clean pages and may be useful for small tables with high intersystem read interest to reduce contention.

UNREGISTER PAGES

Number of coupling facility requests to unregister interest to the GBP for a single page (QBCADG).

This is generally done as DB2 steals pages from the local buffer pool that belong to GBP-dependent page sets or partitions (DB2 5.1 and later).

EXPLICIT X-INVALID

Number of explicit cross-invalidations (QBGAEX).

(DB2 6.1 and later)

WRITE TO SECONDARY

Number of coupling facility requests to write changed pages to the secondary group buffer pool for duplexing (QBGA2W).

(DB2 6.1 and later)

## GLOBAL LOCKING ACTIVITY

This section is available for DB2 4.1 and later only.

PLOCK REQUESTS

-LOCK

Number of lock requests for P-locks (QTGALPLK).

Physical locks (P-locks) are acquired in the DB2 data sharing environment at the data set level to allow DB2 to determine whether an object is of intersystem interest.

Concurrency is still managed as before with transaction (or logical) locks (L-locks).

A P-lock is acquired for the first transaction that accesses an object (table space, index space, partition) but it is owned by the subsystem. Page P-locks are used for row-level locking. P-locks never cause timeouts, as the lock mode can be negotiated. Intersystem interest occurs only when at least one DB2 has update interest and one or more others have read interest.

-CHANGE

Number of change requests for P-locks (QTGACPLK).

This counter is incremented when a P-lock has been established and the IRLM associated with this DB2 determines that the nature of a that lock must be changed for page sets which are of intersystem interest.

The number of P-lock changes reflects the changing status of page sets as access modes change and as the lock state is negotiated between DB2s.

-UNLOCK

Number of unlock requests for P-locks (QTGAUPLK).

This number represents the number of IRLM unlock requests in a data sharing environment.

XES REQUESTS

-LOCK

Number of lock requests propagated to MVS XES synchronously (QTGALSLM).

This number is incremented for all data sharing locks when a request for a lock on an object of DB2 intersystem interest is sent to MVS Cross-System Services. Both L-locks and P-locks are included. The count is not incremented if a suspension occurs.

This number reflects the global lock activity.

-CHANGE

Number of change requests propagated to MVS XES synchronously (QTGACSLM).

This number is incremented for all data sharing locks when a change request for a lock on an object of DB2 intersystem interest is sent to MVS Cross-System Services. Both L-locks and P-locks are included. The count is not incremented if a suspension occurs.

This number reflects the number of change requests for locks of intersystem interest.

-UNLOCK

Number of unlock requests propagated to MVS XES synchronously (QTGAUSLM).

This number is incremented for all data sharing locks when an unlock request for an object of DB2 intersystem interest is sent to MVS Cross-System Services. Both L-locks and P-locks are included. The count is not incremented if a suspension occurs.

This number reflects the number of resources propagated for unlocks, not just the number of unlock requests, which may be generic for a group of resources.

SUSPENDS

IRLM

Number of global IRLM suspensions (QTGAIGLO).

This number is incremented when an incompatible lock is requested on an object (page set, page or row) in a data sharing environment that is being held by another system.

For a complete picture of global contention, all three global suspension counts must be considered.

This counter is a measure of intersystem contention. All locks of intrasystem contention are resolved prior to the IRLM checking for intersystem contention. This number is the number of true waits for another thread on a different MVS that has an incompatible lock type to the requester. Global contention requires intersystem communication to resolve the conflict.

XES

Number of global XES suspensions (QTGASGLO).

This counter is a measure of MVS XES global resource contention.

The MVS XES lock states were in conflict but IRLM lock states were not.

FALSE

Number of false global suspensions (QTGAFLSE).

This counter is incremented every time MVS Cross-System Services detects contention on the resource hash class but not on the resource itself.

**Tuning Tip:** MVS Cross-System Services uses a hash table for efficient determination of whether a resource is locked. This hash table points to a number of synonyms. When contention exists on the hash class but not on the actual object, this is false contention. The more resources declared to be of intersystem interest, the more chances there are of this condition occurring. The CF lock structure may be too small.

INCOMPATIBLE RETAINED LOCK

Number of global requests denied because of incompatible retained lock (QTGADRTA).

This number is incremented every time Global Lock Services denies a lock request because an incompatible lock type has been retained for the requested resources.

This number reflects the instances in which intersystem access to a page or row cannot be obtained because another thread on another system had access to the resource requested but the other system failed. The CF retains locks until the other system has come up to complete the unit of work.

NOTIFY MESSAGES SENT
Number of notify messages sent (QTGANTFY).

This number represents the number of messages passed to the coupling facility; for example, to notify other members that DBDs have been changed due to CREATE, ALTER, or DROP statements.

This is the outbound traffic to the coupling facility in a data sharing complex.

GLOBAL LOCK CONTENTION(%)
Percentage of global lock contentions.

This value is calculated as

Global Suspends / Total Global Locks x 100.0

where

Global Suspends is the total number of global lock contentions (QTGAIGLO + QTGASGLO + QTGAFLSE).

Total Global Locks is the sum of

- Lock requests (QTGALSLM)
- Change requests (QTGACSLM)
- Unlock requests (QTGAUSLM)
- Global lock contentions (QTGAIGLO + QTGASGLO + QTGAFLSE)

FALSE CONTENTION(%)
Percentage of global lock false contentions.

This value is calculated as

False Contentions / Global Suspends x 100.0

where

False Contentions is the total number of global lock suspensions due to false contention (QTGAFLSE).

Global Suspends is the total number of global lock contentions (QTGAIGLO + QTGASGLO + QTGAFLSE).

## TERMINATION REASON

NORMAL

Number of times a thread terminated for normal reasons (QWACRINV):

- NEW USER

A new user signed on.

- RESIGNON

A partial signon occurred where the authorization stayed the same.

This situation occurs primarily when the TOKENE transaction in CICS is set to YES.

- DBAT INACTIVE

A Database Access Thread (DBAT) became inactive.

This situation occurs when an active DDF thread does a commit or rollback or when DB2 inactivates the thread because no work is received for the interval specified in IDLE THREAD TIMEOUT.

- DEALLOCATION

The application informed DB2 that it has terminated.

RRSAF COMMIT

An application using the Recoverable Resource Manager Services Attachment Facility (RRSAF) specified a value of commit for the accounting interval parameter of the signon function (DB2 5.1 and later).

- APPL PROG END

MVS informed DB2 that a thread has been terminated.

- IFI READS

An Instrumentation Facility Interface synchronous read request terminated.

This situation is usually invoked by monitor programs.

ABNORMAL

Number of times a thread terminated for abnormal reasons (QWACRINV):

- APPL PROG ABEND

The application abended.

- END-OF-MEMORY

The application was forced by command or system error and abended.

- RESOLVE INDOUBT

A participant in a two-phase commit operation went through recovery processing (usually automatically or by operator command) and committed or rolled back work.

- CANCEL/FORCE DB2

One of these commands was issued:

- STOP FORCE (from DB2)
- CANCEL DB2 (from MVS)
- FORCE DB2 (from MVS)

INDOUBT

Number of times a thread was left indoubt when it terminated during a two-phase commit operation for these reasons (QWACRINV):

- APPL PROG END

  Normal end of task.

- APPL PROG ABEND

  Abnormal end of task.

- END-OF-MEMORY

  Abnormal end of memory.

- CANCEL/FORCE DB2

  One of these commands was issued:

  - STOP FORCE (from DB2)
  - CANCEL DB2 (from MVS)
  - FORCE DB2 (from MVS)

## ROUTINES

STORED PROCS -CALL STATEMENTS
Number of SQL CALL statements executed (QXCALL).

This is a count of the number of SQL CALL statements processed for the application. This activates a DB2 stored procedure (DB2 4.1 and later).

STORED PROCS -ABENDS
Number of stored procedure abends (QXCALLAB).

This counter is incremented when a stored procedure abends.

**Tuning Tip:** Large numbers here indicate that one or more stored procedures have fatal errors.

STORED PROCS -TIMEOUTS
Number of SQL CALL statements timed out (QXCALLTO).

This counter is incremented when the Stored Procedures Address Space does not have a TCB available within the time limit specified in the DSNZPARM TIMEOUT parameter to schedule the execution of an SQL CALL statement.

**Tuning Tip:** Appropriate fixes include

- Reduce the amount of CPU service units in other stored procedures to get better throughput.
- Stop stored procedures no longer needed.
- Increase the number of TCBs available in the Stored Procedures Address Space (by increasing the NUMTCB parameter in the Stored Procedures Address Space JCL).

STORED PROCS -REJECTS
Number of SQL CALL statements rejected because procedure was stopped (QXCALLRJ).

This counter is incremented when an SQL application CALLs a stored procedure which has been stopped by the operator or system administrator.

**Tuning Tip:** If this problem happens frequently, be sure the procedure is started in normal mode.

USER DEF FUNC -EXECUTED
Number of user-defined functions executed (QXCAUD).

(DB2 6.1 and later)

USER DEF FUNC -ABENDS
Number of times a user-defined function abended (QXCAUDAB).

(DB2 6.1 and later)

USER DEF FUNC -TIMEOUTS
Number of times a user-defined function timed out waiting to be scheduled (QXCAUDTO).

(DB2 6.1 and later)

USER DEF FUNC -REJECTS

Number of times a user-defined function was rejected (QXCAUDRJ).

(DB2 6.1 and later)

TRIGGER-STATEMENT ACTIVATIONS

Number of times a statement trigger is activated (QXSTTRG).

(DB2 6.1 and later)

TRIGGER-ROW ACTIVATIONS

Number of times a row trigger is activated (QXROWTRG).

(DB2 6.1 and later)

TRIGGER-ERRORS

Number of times an SQL error occurred during execution of a triggered action (QXTRGERR).

(DB2 6.1 and later)

MAX NESTED SQL CASCADING

Maximum level of nested SQL cascading due to triggers, user-defined functions, and stored procedures (QXCASCDP).

(DB2 6.1 and later)

## MISCELLANEOUS

ROWID - DIRECT ACCESS
Number of times that DB2 used direct row access to locate a record (QXROIMAT).

(DB2 6.1 and later)

ROWID - INDEX USED
Number of times that DB2 attempted to use direct row access but reverted to using an index to locate a record (QXROIIDX).

(DB2 6.1 and later)

ROWID - TS SCAN USED
Number of times that DB2 attempted to use direct row access but reverted to using a table space scan to locate a record (QXROITS).

(DB2 6.1 and later)

MAX LOB STORAGE
Maximum storage used for LOB values, in megabytes (QXSTLOBV).

(DB2 6.1 and later)

LOG RECS WRITTEN
Number of log records written (QWACLRN).

The amount of logging for a thread can exceed the amount of logging for units of recovery because the logging for a thread includes logging of actions performed by DB2 on behalf of the thread.

(DB2 6.1 and later)

LOG BYTES WRITTEN
Total number of bytes of log records written (QWACLRAB).

(DB2 6.1 and later)

# Accounting Long Report—Packages

The fields in this section are organized in the following categories. For the highlights, see page 130.

## PACKAGE / DBRM ACTIVITY

TYPE (PKG, DBRM, BOTH)
Type of program: DBRM, package, or both. BOTH can appear in reports if there are packages and DBRMs with the same program name.

PACKAGE COLLECTION ID
Package collection ID (QPACCOLN).

This is the name of the collection to which a package belongs. It is blank for a DBRM.

A collection is a logical grouping of packages.

LOCATION NAME
Location name (QPACLOCN).

This is the location name for a package which is remotely bound. If blank, the DBRM or package is executed locally.

**Tuning Tip:** This field is useful for determining whether this package was remotely bound. If the DBRM or package is remotely bound, all times in the package/DBRM accounting are local times to execute the remote package for this requesting thread.

PROGRAM NAME
Program ID (QPACPKID).

This is the program name for a DBRM or package. It correlates to the EXPLAIN program ID.

OCCURRENCES
Number of occurrences of this program.

SQL STATEMENTS COUNT
SQL request count (QPACSQLC).

This number is the total number of SQL statements executed from this package or DBRM, including COMMIT or ROLLBACK statements.

**Tuning Tip:** This field assists you in determining which packages are more SQL intensive. The totals for packages may not equal the totals in the accounting section because of differences in the counters such as commits and rollbacks.

DB2 EXIT/ENTRY EVENTS
Number of exit/entry events (QPACARNA).

This value indicates the number of DB2 entry and exit events processed in determining the elapsed and processor times in DB2.

Accounting class 7 must be activated to determine this value.

TOTAL ELAPSED EXECUTION TIME

Elapsed time of a package/DBRM execution (QPACSCT).

This time represents the total elapsed times of all executions of this package or DBRM. Accounting class 7 must be active to gather this data.

**Tuning Tip:** This information can provide more granular information as to which package or DBRM consumed the most resources.

TOTAL TCB CPU TIME IN PACKAGE

Total CPU time for package/DBRMs (QPACTJST).

This field is the total CPU time for all in-DB2 time spent by all executions of this package or DBRM during the life of the thread. Accounting class 7 must be active to collect this data.

**Tuning Tip:** This number can provide more granularity than plan level accounting to determine where resource overhead is occurring.

CPU IN PARALLEL TASKS

CPU time spent processing parallel tasks (DB2 4.1 and later).

TOTAL WAIT TIME

All class 8 wait times.

CLASS 7 NOT NULL COUNT

Total number of accounting records with class 7 times or events.

SERVICE UNITS - TOTAL

Total CPU time converted to service units using the conversion factor from QWACSUCV (DB2 5.1 and later) as follows:

SU = CPU seconds x (16000000/conversion factor)

SERVICE UNITS - PARALLEL TASKS

CPU time spent processing parallel tasks converted to service units using the conversion factor from QWACSUCV (DB2 5.1 and later) as follows:

SU = CPU seconds x (16000000/conversion factor)

## PKG./ DBRM WAIT TIMES

LOCK/LATCH
Wait time for lock/latch (QPACAWTL).

This value indicates the elapsed time the thread waited for transaction locks and latches while executing this package/DBRM. Accounting class 8 must be activated to determine this value.

**Tuning Tip:** This time reflects the amount of time the thread was suspended because of an incompatible lock or latch mode.

SYNC I/O
Wait time for synchronous I/O (QPACAWTI).

This time is the elapsed time spent waiting for all synchronous I/O while executing this package/DBRM. Accounting class 8 must be activated to determine this value.

**Tuning Tip:** Most of the time included in this time is synchronous read time, but under stress conditions it can reflect synchronous write time (when the immediate write threshold is reached) or wait for synchronous log I/O.

OTHER THREAD READ I/O
Wait time for asynchronous reads (QPACAWTR).

This value is the time spent waiting for reads done under another thread to complete while executing this package/DBRM. These waits may be for sequential prefetch, sequential detection, list prefetch, or for synchronous I/O done while executing a sequential prefetch operation.

Accounting class 8 must be activated to collect this number.

**Tuning Tip:** Time reflected here does not reflect the duration of the I/O but simply the time the thread waited for completion of the I/O activity. If this number is high, parallel processing may assist in reducing this time.

OTHER THREAD WRITE I/O
Wait time for write I/O (QPACAWTW).

This value is the time spent waiting for an asynchronous write I/O to complete or for a synchronous write I/O to be performed for another thread while executing this package/DBRM. Accounting class 8 must be activated to obtain this time.

**Tuning Tip:** Time reflected here should be small, since normal checkpoint and deferred write thresholds protect threads from waiting for asynchronous write time. If this number is large, the buffer pool size or deferred write threshold may need to be adjusted.

EXECUTION TASK SWITCH

Wait time for DB2 services (QPACAWTE).

This value is the time spent waiting for special DB2 services while executing this package/DBRM. A synchronous execution unit switch is done from the thread to one of the DB2 service tasks.

Service times reported here include

- Open/close of a page set
- Update to SYSLGRNG or SYSLGRNX directory page set
- Commit phase 2 for read-only threads originating from TSO or batch
- Time to recall a page set from HSM (or similar) archive device
- Dataspace manager services
- Define data set (for CREATE operations)
- Extend data set (to take secondary extents)
- Delete data set (for DROP operations)

Accounting class 8 must be activated to obtain this time.

ARCHIVE LOG QUIESCE

Wait time for log quiesce (QPACALOG).

This value is the time spent waiting for an -ARCHIVE LOG MODE(QUIESCE) to take place while executing this package/DBRM.

This process externalizes buffers and switches the active log data set. Time reflected here is the time the thread waited, not the time for the command to complete. Accounting class 8 must be activated to obtain this time.

ARCHIVE LOG READ

Wait time for archive read from tape (QPACAWAR).

This is the amount of time spent waiting for reads from an archive tape while executing this package/DBRM.

Time spent waiting for an archive tape is either for the completion of the RECOVER utility or for an application program, which has not committed, to read the archive tape in order to complete backout. Accounting class 8 must be activated to obtain this time.

**Tuning Tip:** If this time is large, the application program should commit more often. It may also make sense to increase the size and number of active logs.

DRAIN LOCK

Wait time for drain lock (QPACAWDR).

This value is the time spent by a requester of a drain lock, such as a utility or command, waiting to acquire that lock while executing this package/DBRM.

Processes, such as utilities that serialize events to page sets, must acquire a drain lock. This is the time spent waiting for this event. Accounting class 8 must be activated to obtain this time.

CLAIM RELEASE

Wait time for claim release (QPACAWCL).

This value is the time spent by the holder of a drain lock waiting for SQL claimers to complete their activity while executing this package/DBRM.

Drainers must wait until the SQL claim count reaches zero. This is the time spent waiting for them to clear. Accounting class 8 must be activated to obtain this time.

PAGE LATCH CONTENTION

Wait time for page latch contention (QPACAWTP).

This is the time spent waiting for page latching activities due to lock avoidance while executing this package/DBRM.

Accounting class 8 must be activated to obtain this time.

**Tuning Tip:** Page latching can be avoided by scheduling applications when there is a lower likelihood of latch contention.

STORED PROCEDURE

Wait time for SPAS TCB scheduling (QPACCAST).

This time is the elapsed time spent waiting for an available TCB in the Stored Procedures Address Space to become available for scheduling a stored procedure. It is calculated only if accounting class 8 is active.

**Tuning Tip:** This time should be minimal depending on the workload to the Stored Procedures Address space.

NOTIFY MESSAGES

Wait time for data sharing notify messages (QPACAWTG).

This value is the amount of wait time for sending notify messages to another data sharing member while executing this package/DBRM.

When tables in a data sharing group are CREATed, ALTERed, or DROPped in a data sharing group, intersystem messages must be sent to update the database descriptors (DBDs) in the other members' EDM pools. This is the time spent waiting for this message traffic to occur. This time is collected only when accounting class 8 is activated and data sharing is enabled.

GLOBAL CONTENTION

Wait time for global lock contention (QPACAWTJ).

This time is the amount of elapsed time caused by an IRLM lock suspension due to global lock contention while executing this package/DBRM. It requires intersystem communication to resolve.

This time is collected only when accounting class 8 is activated and data sharing is enabled.

CLASS 8 SUSPENSION TIME

Total of class 8 wait times.

CLASS 8 NOT NULL COUNT

Total number of accounting records with class 8 times or events.

# Accounting Long Report—DDF

The fields in this section are organized in the following categories. For the highlights, see page 130.

**Note:** This page is produced only if distributed work is done.

## DISTRIBUTED ACTIVITY

REMOTE LOCATION

Location name of remote site (QLACLOCN).

Location name of a remote site involved in this distributed application, either as a server or a requester.

PRODUCT ID

Remote product ID (QLACPRID).

Product ID of the remote location.

The product ID begins with DSN for DB2, ARI for SQL/DS, SQL for client/server, or QSQ for DB2/400. This is followed by the version, release, and modification level, in the format vvrrm.

METHOD

Method of access used: APPL for application-directed, SYST for system-directed, or BOTH.

DDF ACCESSES

Number of occurrences of activity with this remote location and the local location.

TRANSACTIONS SENT

Number of transactions sent (QLACTRNS).

This counter is incremented at the requester location and reflects the total number of threads requested at the server location. Each one allocates a DBAT thread.

This data is useful only at the requester location since DB2 maintains local site autonomy based on the number of active and inactive threads in DSNZPARM.

TRANSACTIONS RECEIVED

Number of transactions received (QLACTRNR).

This counter is incremented at the server location and reflects the number of threads requested by the requester.

This counter is useful only at the server location since DB2 maintains local site autonomy based on the number of active and inactive threads in DSNZPARM.

MESSAGES SENT

Number of messages sent to the remote location (QLACMSGS).

This counter is incremented at the originating location. It reflects a count of each VTAM message.

A message, in VTAM terms, is a group of characters and control bit sequences which are transmitted as a single unit of data, including header information and data.

MESSAGES RECEIVED
Number of messages received from the remote location (QLACMSGR).

This counter is incremented at the receiving locations reflecting the number of messages this location received.

The count of messages received can often be more than the number of messages sent since VTAM adds control header information as part of its protocols.

BYTES SENT
Number of bytes sent to the requester (QLACBYTS).

This count represents the number of bytes sent by the server. This counter is incremented at the server location.

The number of bytes per message can help the network tuner decide the appropriate RUSIZE and pacing to fit the network.

BYTES RECEIVED
Number of bytes received from the server (QLACBYTR).

This counter is incremented at the requester reflecting the number of bytes received by the server.

The number of bytes received may be greater than the number sent since the overhead of network transmission tends to add message information in the form of additional VTAM header blocks.

ROWS IN BUFFER
Number of rows transmitted or received using block fetch (QLACBROW).

This number, incremented at both requester and server, indicates the number of rows transmitted by block fetch operations.

**Tuning Tip:** Generally speaking, the higher the number of rows transmitted by block fetch, the better performance for distributed transactions will be. At the requester location, compare this value to the number of rows received to determine what percentage of the rows were returned from the server with block fetch. For a server location, compare it to the number of rows sent.

CONVERSATIONS SENT
Number of conversations initiated from the requester location (QLACCNVS).

This counter is incremented at the requester location and reflects the number of specific conversations sent to all server locations.

**Tuning Tip:** Conversations are specific instances of transmitting data back and forth. Sessions are logical connections between requesters and servers. It is analogous to a telephone call in which the call is a session and each party talking is a conversation.

CONVERSATIONS RECEIVED
Number of conversations initiated from the requester (QLACCNVR).

This counter is incremented at the server location for each conversation the server has successfully retrieved.

**Tuning Tip:** Performance tuning is the art of getting as many data rows possible in as few conversations as possible.

CONVERSATIONS ALLOCATED
Number of successful conversation allocations (QLACCNVA).

This value is incremented at the requester location and includes only successful conversation allocations.

CONVERSATIONS TERMINATED
Number of conversations terminated (QLACCNVT).

This value is incremented at the requester location and includes all conversations terminated.

This value might not be the same as the number allocated because not all conversations are terminated when the accounting record is written.

CONVERSATIONS QUEUED
Number of conversations queued (QLACCNVQ).

This counter is incremented at the requester location every time a conversation request to VTAM is initiated but cannot yet be started.

**Tuning Tip:** The way to avoid queuing conversational requests from DB2 is to increase the conversation limits column in the SYSIBM.SYSLMODE table to a larger number. This should be done in conjunction with the VTAM system programmer to ascertain whether the network can handle increased numbers of conversations.

MAX CONVERSATIONS OPEN
Largest number of conversations open (QLACCIEL).

This value is incremented at the requester location when a new high-water mark is reached.

## SERVER

SQL STATEMENTS SENT

Number of SQL statements sent to the remote server (QLACSQLS).

This counter is incremented at the requester location when an SQL statement is sent.

**Tuning Tip:** This information, along with other message information, can help determine the relative efficiency of the distributed applications and can, in conjunction with other data items, help in the network tuning of distributed applications.

ROWS RECEIVED

Number of rows received from the server (QLACROWR).

This counter is incremented at the requester location and reflects the number of rows retrieved from the server.

**Tuning Tip:** From a performance perspective, the ideal is to retrieve as many rows in as few conversational messages as possible. This is accomplished using a feature of DB2 called block fetch. Block fetch is used when a cursor is determined unambiguously to be a read only cursor. The use of the FOR FETCH ONLY clause along with the bind option of CURRENTDATA(NO) can help DB2 achieve this goal. Block fetch will always be most effective with system-directed SQL (three-part names without a CONNECT statement, sometimes called DB2 private protocols). In using private protocols, DB2 will fit as many rows as possible into a single 32K message. The typical overhead in distributed performance is network bandwidth. If the application can send fewer messages, the distributed application will be more efficient. For further information, consult the Performance, Monitoring and Tuning section of the *IBM DATABASE 2 Administration Guide*.

BLOCKS RECD USING BLOCK FETCH

Number of blocks received using block fetch (QLACBRBF).

This counter, incremented at the requester location, indicates how many blocks were received to process remote requesters.

**Tuning Tip:** Each block costs VTAM transmission time. The more rows that can be transmitted per block, the better DB2 performance is for distributed transactions.

SWITCH TO LIMITED BLOCK FETCH

Number of switches from continuous to limited block fetch (QLACCBLB).

This counter is incremented at the requester location and applies only to system-directed applications (three-part names, no CONNECT statement, sometimes called private protocols). It reflects the number of times asynchronous block fetch had to be reduced to a more limited form of block fetch because of a lack of network resources.

**Tuning Tip:** This situation occurs when too many conversations are occurring at this site for VTAM to handle the asynchronous fetch of data. It needs to be resolved by the MVS and VTAM systems programmers to allow more conversations to take place.

SQL BOUND FOR REMOTE ACCESS

Number of SQL statements bound for remote access (QLACRBND).

This value is incremented at the requester location each time a system-directed SQL statement is encountered (three-part name, no CONNECT statement, DB2 private protocols where dynamic requests occur at the server).

**Tuning Tip:** One of the expensive features of DB2 private protocol is that the SQL statement is dynamically run on the server site and thus incurs extra overhead at the server.

1PHASE COMMITS SENT

Number of commit requests sent to the server (QLACCOMS).

This counter is incremented at the requester and reflects the number of single-phase commits sent (TSO, Batch, CAF).

Two-phase commit operations (updating distributed transactions, CICS, and IMS) are not included.

1PHASE ROLLBACKS SENT

Number of rollback requests sent to the server (QLACABRS).

This counter is incremented at the requester location and reflects the number of single-phase rollback requests.

Two-phase commit operations (updating distributed transactions, CICS, and IMS) are not included.

2PHASE COMMITS SENT

Number of two-phase commits sent to the participant (QLACCRSE).

This counter is incremented by the coordinator each time a phase 1 request is sent.

In a two-phase commit operation, the coordinator then sends a phase 2 request to all participants involved in an update. Assuming all vote YES, the participants and coordinator release locks and issue a *forget* message to all logs.

2PHASE BACKOUTS SENT

Number of backout requests sent to the participant (QLACBKSE).

This counter is incremented by the coordinator when a NO vote has been received at the end of phase 1 or a transaction has abended.

Backout processing will cause all participants and coordinator to backout to the last commit.

2PHASE PREPARES SENT

Number of prepare requests sent to the participant (QLACPRSE).

This counter is incremented by the coordinator to all participants in DB2 two-phase commit protocols.

In distributed processing, two-phase commit works similarly to two-phase commit on a local system from CICS or IMS. A coordinator (in this case the server) tells all participating DB2 subsystems to issue a phase 1 prepare to commit. The transaction completes when all participants indicate they have logged all data and phase 2 has been accomplished.

2PHASE LAST AGENT SENT

Number of last agent requests sent to the coordinator (QLACLASE).

When DB2 is the requester to another DB2, this counter is incremented by the participant each time a conversation is deallocated and it was not used since the last commit. This applies only to two-phase commit processing.

**Tuning Tip:** The advantage is that this permits DB2 to save message overhead to acknowledge the commit request. If the application program issues a RELEASE SQL verb before the commit, another message can be saved.

The art of VTAM distributed performance tuning is to reduce the number of messages sent or received. By taking advantage of last agent requests and sensible use of the RELEASE SQL verb, the application program can help save message traffic across a distributed transaction.

2PHASE FORGET RESPONSES RECD

Number of two-phase commits received from the coordinator (QLACCRRC).

This counter is incremented by the participant each time a phase 1 request is received.

In a two-phase commit operation, the coordinator then sends a phase 2 request to all participants involved in an update. Assuming all vote YES, the participants and coordinator release locks and issue a *forget* message to all logs.

2PHASE COMMIT RESPONSES RECD

Number of request commits received by the coordinator (QLACVYRC).

This counter is incremented by a coordinator when it has received request commits from all participants involved in the update at the end of phase 1.

This counter is incremented when the participant has been involved in an update; otherwise a forget record would have been received.

2PHASE BACKOUT RESPONSES RECD

Number of backout requests received by the coordinator (QLACVNRC).

This counter is incremented by a coordinator in a two-phase commit operation that has one or more participants voting NO.

With normal processing, the coordinator issues a backout request to all participants after this occurs.

ELAPSED TIME AT REQUESTER

Elapsed time at requester waiting for DBAT (QLACCPUL).

Elapsed time at the requester location spent waiting for completion of work by a database access agent.

This value is maintained at the requester location. It includes both DB2 and VTAM processing time as well as network time. It is calculated by accumulating the difference between the store clock value obtained before and after each VTAM request.

ELAPSED TIME AT SERVER

Elapsed DBAT server time (QLACCPUR).

Elapsed database access agent time at the serving location. This value is updated at the requester location and is reported only for DB2 private protocol.

If the agent uses both system-directed and application-directed access, this time can be misleading. The value is sent from the server location to the requester location and does not includes VTAM or network elapsed time. The value represents the elapsed time the DBAT at the server spent satisfying the requests. When block fetch is used, this time can be larger than the elapsed time spent waiting at the requester (QLACCPUL).

**Tuning Tip:** Compare this value with the accounting class 2 time (allied agent time in DB2) to see if the distributed allied thread that used the DBAT is spending too much time in remote processing.

CPU TIME AT SERVER

DBAT server CPU time (QLACDBAT).

Database access agent CPU time at the serving location. This value is updated at the requester location and is intended for problem determination only.

The reported value does not include most of the time spent in VTAM and is reported only for DB2 private protocol access requests. If the agent uses both DB2 private protocol and DRDA protocol, this value can be misleading. The value represents the CPU time the DBAT at the server spent satisfying the requests. This value is not always updated at the requester location, but the actual CPU usage can be seen at the server location.

## REQUESTER

SQL STATEMENTS RECEIVED

Number of SQL statements received from the requester (QLACSQLR).

This counter is incremented at the server location when an SQL statement is received.

**Tuning Tip:** This information, along with other message information, can help determine the relative efficiency of the distributed applications and can, in conjunction with other data items, help in the network tuning of distributed applications.

ROWS SENT

Number of rows sent to the requester location (QLACROWS).

This counter is incremented at the server location for each row sent to a remote requester location.

**Tuning Tip:** This number, combined with the number of messages (conversations), can be used in determining whether efficiency techniques such as block fetch are being utilized.

BLOCKS SENT USING BLOCK FETCH

Number of blocks transmitted using block fetch (QLACBTBF).

This counter, incremented at the server location, indicates how many blocks were sent to remote requesters.

**Tuning Tip:** Each block costs VTAM transmission time. The more rows that can be transmitted per block, the better DB2 performance is for distributed transactions.

1PHASE COMMITS RECEIVED

Number of commit requests received from the requester (QLACCOMR).

This counter is incremented at the server location for single-phase commits only.

Two-phase commit operations (updating distributed transactions, CICS, and IMS) are not included.

1PHASE ROLLBACKS RECEIVED

Number of rollback requests received from the sender (QLACABRR).

This counter is incremented at the server location reflecting the number of single-phase rollback requests received.

Two-phase commit operations (updating distributed transactions, CICS, and IMS) are not included.

2PHASE COMMITS RECEIVED

Number of two-phase commits received from the coordinator (QLACRRRC).

This counter is incremented by the participant each time a phase 1 request is received.

In a two-phase commit operation, the coordinator then sends a phase 2 request to all participants involved in an update. Assuming all vote YES, the participants and coordinator release locks and issue a *forget* message to all logs.

2PHASE BACKOUTS RECEIVED

Number of backout requests received by the participant (QLACBKRC).

This counter is incremented when the participating DB2 receives a backout request from the coordinator. This usually occurs when the coordinator has received a NO vote from phase 1 processing or an abend has occurred.

Backout processing will cause all participants and coordinator to backout to the last commit.

2PHASE PREPARES RECEIVED

Number of prepare requests received by the participant (QLACPRRC).

This counter is incremented by the participant (usually the requester) each time the coordinator issues the prepare to all participants in DB2 two-phase commit protocols.

In distributed processing, two-phase commit works similarly to two-phase commit on a local system from CICS or IMS. A coordinator DB2 tells all participating DB2 subsystems to issue a phase 1 prepare to commit. The transaction completes when all participants indicate they have logged all data and phase 2 has been accomplished.

2PHASE LAST AGENT RECEIVED

Number of last agent requests received (QLACLARC).

When DB2 is the coordinator to another DB2, this counter is incremented by the coordinator each time a) a conversation is deallocated and b) the conversation was not used since the last commit. This applies only to two-phase commit processing. The advantage is that this permits DB2 to save message overhead to acknowledge the commit request. If the application program issues a RELEASE SQL verb BEFORE the commit, another message can be saved.

**Tuning Tip:** The art of VTAM distributed performance tuning is to reduce the number of messages sent or received. By taking advantage of last agent requests and sensible use of the RELEASE SQL verb, the application program can help save message traffic across a distributed transaction.

2PHASE FORGET RESPONSES SENT

Number of forget responses sent to the coordinator (QLACRRSE).

This situation occurs and a counter is incremented by the participant in a two-phase commit operation when the participant does not apply any updates.

The forget response takes the participant out of the voting chain when requested to prepare for commit.

2PHASE COMMIT RESPONSES SENT

Number of request commits sent to the coordinator (QLACVYSE).

This counter is incremented by a participant when it has completed phase 1 processing and is ready to perform phase 2 processing.

This counter is incremented only when the participant has been involved in an update in this unit of work.

2PHASE BACKOUT RESPONSES SENT

Number of backout requests sent to the coordinator (QLACVNSE).

This counter is incremented by a participant in a two-phase commit operation that has voted NO because something has failed during phase 1 processing.

NO votes usually correlate to abends at the requester.

REMOTE COMMITS PERFORMED

Number of commit operations performed with remote location as the coordinator (QLACCPTR).

This counter is incremented when a participant successfully completes the phase two process.

This indicates that two phase processing has resulted in a successful commit.

REMOTE ROLLBACKS PERFORMED

Number of rollback operations performed with remote location as the coordinator (QLACRBTR).

This counter is incremented when a participant has received a ROLLBACK request from the coordinator.

**Tuning Tip:** This indicates that one or more participants or coordinator received a NO vote to commit. High numbers indicate problems with the application in one or more sites.

REMOTE INDOUBT THREADS

Number of threads that became indoubt with the remote location as the coordinator (QLACINDT).

An indoubt situation can exist during phase 2 processing after receipt of prepare for phase 2 but before response to end of phase 2. This indicates that communication with the remote coordinator was lost.

**Tuning Tip:** Indoubt threads will resolve themselves automatically upon successful restart of the failing participant or coordinator and the network path used to send the messages. The problem occurs when the systems are intact but communication has been severed. This can cause indoubt threads to be maintained for long periods of time. A -RESOLVE INDOUBT command may be issued to free these locks but ***only*** after a responsible system programmer has determined the correct course of action: COMMIT or ABORT.

# Chapter 9. Statistics Short Report

This chapter describes the statistics short report, as shown in Figure 20–Figure 22. It has two pages per summary interval, followed by a page for DDF information, if present.

```
LOCATION: DB2J               GROUP : DBGJ                 MAINVIEW FOR DB2                  INTERVAL START: 2001-01-27-02.12.44.0000
SSID    : DB2J               MEMBER: DB2J                PERFORMANCE REPORTER               INTERVAL END  : 2001-03-02-16.59.30.0000
VERSION : 71                 SCOPE : MEMBER             STATISTICS SHORT REPORT             INTERVAL      : 00001     PAGE 0001/0002
                                                                                                            RECORDS : 00226

----HIGHLIGHTS 1----------------  -----QTY   ----HIGHLIGHTS 2----------------  -----QTY   ----HIGHLIGHTS 3----------------  -----QTY
CREATE THREADS                        430    COMMITS                               766    SYNC COMMITS                          676
DISTRIBUTED THREAD QUEUED               0    COMMITS - PHASE 2                       0    ROLLBACK                               90
INCREMENTAL BINDS                       0    COMMITS - READ ONLY                     0

----CPU TIMES (SECONDS)---------            ---TOTAL            -----TCB            -----SRB
CPU TIME FOR MSTR                            1.5162K             1.3394K           176.8368
CPU TIME FOR DBM                           908.8873            198.0736            710.8137
CPU TIME FOR IRLM                          360.5312              3.4340            357.0972
CPU TIME FOR DIST                           52.0636             48.2990              3.7646

----SQL DML---------------------  -----QTY   ----SQL DCL---------------------  -----QTY   ----SQL DDL---------------------  -----QTY
SELECT                                  0    LOCK TABLE                              0    CREATE                                107
INSERT                                 18    GRANT                                  21    DROP                                   18
UPDATE                                 50    REVOKE                                  0    ALTER                                  24
DELETE                                 38    SET CURRENT SQLID                       0    TOTAL                                 149
PREPARE                               320    SET HOST VARIABLE                       0
DESCRIBE                               17    SET CURRENT DEGREE                      0    COMMENT ON                              0
DESCRIBE TABLE                          0    SET CURRENT RULES                       0    RENAME TABLE                            0
OPEN                                  196    SET CURRENT PATH                        0    LABEL ON                                0
FETCH                               75352    CONNECT TYPE 1                          0
CLOSE                                 193    CONNECT TYPE 2                          6    ----RID LIST--------------------  -----QTY
TOTAL                               76184    SET CONNECTION                          0    RID CURRENT BLOCKS ALLOCATED            0
                                             RELEASE                                 0    RID HIGH BLOCKS ALLOCATED               0
----EDM POOL--------------------  -----QTY   CALL                                    0    RID TERM OVER RDS LIMIT                 0
PAGES IN EDM POOL                    7343    ASSOCIATE LOCATOR                       0    RID TERM TOO MANY CONCUR                0
FREE PAGES                           7321    HOLD LOCATOR                            0    RID TERM - NO STORAGE                   0
EDM POOL FULL FAILURES                  0    FREE LOCATOR                            0    RID TERM - OVER DM LIMIT                0
CT PAGES USED                           1    ALLOCATE CURSOR                         0
DBD PAGES USED                         64    TOTALS                                 54
PKG TBL PAGES USED                      1
SKCT PAGES USED                         6    ----LOG ACTIVITY----------------  -----QTY   ----LOCKING ACTIVITY------------  -----QTY
SKPT PAGES USED                         3    LOG READS FROM BUFFER                  93    TIMEOUT                                19
CT REQUESTS                           162    LOG READS FROM ACTIVE               93204    DEADLOCK                                7
CT NOT IN EDM POOL                     28    LOG READS FROM ARCHIVE                  0    SUSPEND - LOCK CONFLICT               166
CT REQUESTS/CT NOT IN POOL              5    READ DELAY/UNAVAIL RESOURCE             0    SUSPEND FOR LATCH                     294
DBD REQUESTS                         2082    READ ACCESS DELAY-TAPE CONT             0    SUSPEND OTHER REASONS               14346
DBD NOT IN EDM POOL                    52    ACTIVE LOG WRITE REQUESTS          116.48K   LOCK ESCALATION TO SHARED               0
DBD REQ/DBD NOT IN POOL                40    ACTIVE LOG WRITE NOWAIT             60524    LOCK ESCALATION TO EXCLUSIVE            0
PKG TBL REQUESTS                      117    BSDS REQUESTS                         782    DRAIN REQUEST                         809
PKG TBL NOT IN EDM POOL                16    ACTIVE LOG UNAVAIL BUFFER               0    UNSUCCESSFUL DRAIN REQUEST              2
PKG TBL REQ/PKG NOT IN POOL             7    ACTIVE LOG CIS CREATED               5951    CLAIM REQUEST                       19690
PREP STMT CACHE-INSERTS               219    CONTROL INTERVALS OFFLOADED             0    UNSUCCESSFUL CLAIM REQUEST              0
               -REQUESTS              340    ARCHIVE LOG RD ALLOCATIONS              0    LOCK REQUEST                       589.40K
               -PAGES USED           1249    ARCHIVE LOG WR ALLOCATIONS              0    UNLOCK REQUEST                     557.81K
EDM DATA SPACE-PAGES                10240    LOOK-AHEAD TAPE MOUNT ATT.              0    LOCK QUERY REQUEST                    145
              -FREE PAGES           10240    LOOK-AHEAD TAPE MOUNT SUCC              0    LOCK CHANGE REQUEST                  5352
              -FAILURES                 0    LOG SUSPENDS FOR LOG WRITES             0    LOCK OTHER IRLM REQUEST                29
                                             LOG WRITE I/O REQUESTS                  0
                                             LOG CI WRITTEN                          0
                                             LOG SERIAL WRITES                       0
                                             LOG ASYNC WRITES-THRESHOLD              0
                                             LOG BUFFER PAGEDINS                     0
```

Figure 20. Statistics Short Report—Page 1

For a complete description of each section, see

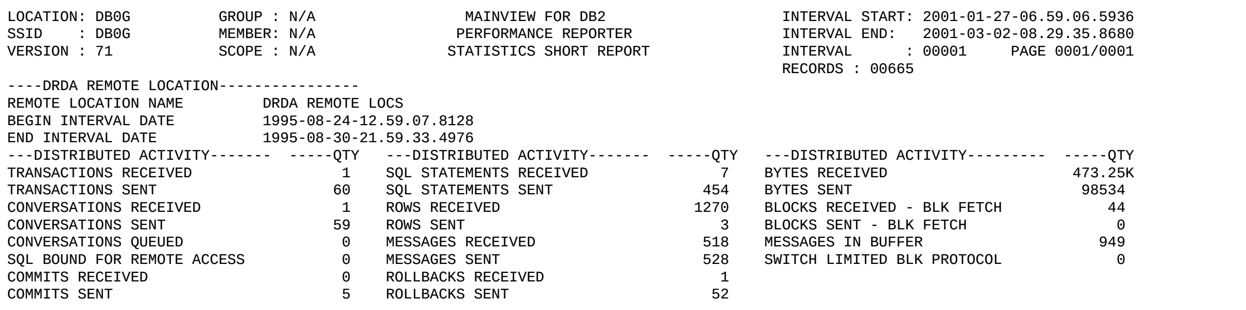
- “HIGHLIGHTS” on page 194
- “CPU TIMES (SECONDS)” on page 196
- “SQL DML” on page 197
- “SQL DCL” on page 198
- “SQL DDL” on page 201
- “RID LIST” on page 202
- “EDM POOL” on page 206
- “LOG ACTIVITY” on page 210
- “LOCKING ACTIVITY” on page 203

```
LOCATION: DB2J               GROUP : DBGJ                  MAINVIEW FOR DB2                   INTERVAL START: 2001-01-27-02.12.44.0000
SSID    : DB2J               MEMBER: DB2J                 PERFORMANCE REPORTER                INTERVAL END  : 2001-03-02-16.59.30.0000
VERSION : 71                 SCOPE : MEMBER              STATISTICS SHORT REPORT              INTERVAL      : 00001     PAGE 0002/0002
                                                                                                            RECORDS : 00226

----HIGHLIGHTS 1----------------  -----QTY   ----HIGHLIGHTS 2----------------  -----QTY   ----HIGHLIGHTS 3----------------  -----QTY
CREATE THREADS                          430    COMMITS                               766    SYNC COMMITS                          676
DISTRIBUTED THREAD QUEUED                 0    COMMITS - PHASE 2                       0    ROLLBACK                               90
INCREMENTAL BINDS                         0    COMMITS - READ ONLY                     0

----BP TOTALS-------------------  -----QTY   ----GLOBAL LOCKING ACTIVITY-----  -----QTY   ----GLOBAL BP TOTALS------------  -----QTY
GETPAGES                          6511.14K   PLOCK REQUESTS-LOCK                  5001    SYNC RD INV BUFF -WITH DATA            58
GETPAGE SEQ REQ                   6392.70K                 -CHANGE                 958               - NO DATA                 12
SYNC READ I/O                         15545                -UNLOCK                1224    SYNC RD NOT FOUND-WITH DATA            18
SYNC READ IO SEQ REQ                   3067    XES SYNC REQ  -LOCK              444.65K              - NO DATA                128
GETPAGES / READ IO                      418                -CHANGE                 690    ASYNC READ - DATA RETURNED              0
SEQ PREFETCH REQ                      23732                -UNLOCK              448.52K              - NO DATA                196
SEQ PREFETCH READ IO                  23318    SUSPENDS      -IRLM                5494    SYNC PAGES WRITTEN -CHANGED          1158
SEQ PREFETCH PAGES READ             670.54K                -XES                    139                       -CLEAN             0
LIST PREFETCH REQUESTS                    0                -FALSE                 2356    ASYNC PAGES WRITTEN-CHANGED           205
LIST PREFETCH READ IO                     0    GLOBAL LOCK CONTENTION(%)               0                       -CLEAN             0
LIST PREFETCH PAGES READ                  0    FALSE CONTENTION(%)                    29    CASTOUT -PAGES WRITTEN                783
DYNAMIC PREFETCH REQUESTS                78    INCOMPATIBLE RETAINED LOCK              0            -NO ENGINE                     0
DYNAMIC PREFETCH READ IO                 30    NOTIFY MESSAGES -SENT               20522            -CLASS THRESHOLD              10
DYNAMIC PREFETCH PAGES READ             662    NOTIFY MESSAGES -RECEIVED            1986            -GBP THRESHOLD                 4
PREFETCH QTY CUT TO 1/2                 133    XES ASYNC REQ -RESOURCES               14    NO WRITE ENGINE                         0
PAGE UPDATES                          50042    NEGOTIATE -PGSET/PARTITION            466    READ FAIL  -STORAGE                     0
PG UPDATE / PG WRITTEN                    5              -PAGE PLOCK                  38    WRITE FAIL -STORAGE                     0
SYNC WRITES                             252              -OTHER PLOCK                132    RD STG STATS(5.1)/OTHER(4.1)            0
ASYNC WRITE IO                          869              -PLOCK CHANGE REQ           512    GBP CHECKPOINT                        420
PAGES WRITTEN / IO                        8    PLOCK/NOTIFY -MAX ENGINES             500    GBP REBUILD                             0
HORIZONTAL DEF. WR REACHED                0                 -NO ENGINE                 0    UNLOCK CASTOUT                        325
VERTICAL DEFER WR REACHED                68                                                 READ CASTOUT CLASS                    576
DM CRITICAL REACHED                       0    ----QUERY PARALLELSIM-----------  -----QTY   READ CASTOUT STATISTICS               425
SUCCESSFUL OPEN                        2860    MAX DEGREE OF PARALLELISM              21    DELETE DIR/DATA ENTRIES              1196
BPOOL HIT RATIO-ALL(%)                   89    PARALLEL GROUPS EXECUTED                6    READ DIRECTORY INFORMATION              0
BPOOL HIT RATIO-RANDOM(%)                89    PLANNED DEGREE                          4    REGISTER PAGE                         884
                                               REDUCED DEGREE - NO BUFFER              1    UNREGISTER PAGE                       189
----SYSTEM SERVICES-------------  -----QTY   FALLBACK - NO BUFFER                    0    REGISTER PAGE LIST                    118
IDENTIFY REQUEST                        320    FALLBACK - NO ESA                       0    REGISTER PAGE LIST-RD CHNGE             0
CREATE THREAD                           430    FALLBACK - CURSOR                       1    REGISTER PAGE LIST-RD CLEAN             0
CREATE THREAD THAT WAITED                 0    FALLBACK - ENCLAVE                      0    EXPLICIT CROSS INVALIDATES              0
SIGNON                                    0    PRLL GROUPS INTENDED                    1    DUPLEX-WRITE REQUEST                    0
TERMINATE                               712    SINGLE DB2-COORDINATOR=NO               0          -WRITE FAIL                       0
ROLLBACK                                 90              -ISOLATION=RR                 0          -DELETE NAME LIST                 0
PREPARE TO COMMIT (PHASE 1)               0    DB2 MEMBER SKIPPED-NO BUF               3          -DELETE NAME                      0
COMMIT (PHASE 2)                          0    REFORMULATED GROUPS-CONFIG              0          -READ CASTOUT STATS               0
READ-ONLY COMMIT                          0                       -NO BUFFER           0
SYNCH COMMIT (SINGLE PHASE)             676                                                 ----ROUTINES--------------------  -----QTY
INDOUBT                                   0    ----OPEN/CLOSE ACTIVITY---------  -----QTY   SPROC-CALL STATEMENTS                   0
INDOUBT RESOLVED                          0    DATASETS CURRENTLY OPEN               543         -ABENDS                            0
END OF MEMORY                             1    MAX D.S. OPEN CONCURRENTLY            543         -TIMEOUTS                          0
END OF TASK                              62    REOPEN FROM SLOW CLOSE              14489         -REJECTS                           0
DB2 CHECKPOINT                           16                                                 TRIGGER-STMT ACTIVATED                  0
                                                                                                   -ROW  ACTIVATED                  0
                                                                                                   -SQL ERRORS                      0
                                                                                            UDF  -EXECUTIONS                        0
                                                                                                 -ABENDS                            0
                                                                                                 -TIMEOUTS                          0
                                                                                                 -REJECTS                           0
```

Figure 21. Statistics Short Report—Page 2

For a complete description of each section, see

- “BP TOTALS” on page 214
- “LOG ACTIVITY” on page 210
- “GLOBAL LOCKING ACTIVITY” on page 223
- “QUERY PARALLELISM” on page 229
- “OPEN/CLOSE ACTIVITY” on page 232
- “GLOBAL BP TOTALS” on page 233
- “ROUTINES” on page 240



```
LOCATION: DB0G          GROUP : N/A                 MAINVIEW FOR DB2                     INTERVAL START: 2001-01-27-06.59.06.5936
SSID    : DB0G          MEMBER: N/A                PERFORMANCE REPORTER                  INTERVAL END:   2001-03-02-08.29.35.8680
VERSION : 71            SCOPE : N/A               STATISTICS SHORT REPORT                INTERVAL      : 00001     PAGE 0001/0001
                                                                                        RECORDS : 00665
----DRDA REMOTE LOCATION----------------
REMOTE LOCATION NAME         DRDA REMOTE LOCS
BEGIN INTERVAL DATE          1995-08-24-12.59.07.8128
END INTERVAL DATE            1995-08-30-21.59.33.4976
---DISTRIBUTED ACTIVITY-------  -----QTY   ---DISTRIBUTED ACTIVITY-------  -----QTY   ---DISTRIBUTED ACTIVITY---------  -----QTY
TRANSACTIONS RECEIVED                 1    SQL STATEMENTS RECEIVED               7    BYTES RECEIVED                     473.25K
TRANSACTIONS SENT                    60    SQL STATEMENTS SENT                 454    BYTES SENT                          98534
CONVERSATIONS RECEIVED                1    ROWS RECEIVED                      1270    BLOCKS RECEIVED - BLK FETCH            44
CONVERSATIONS SENT                   59    ROWS SENT                             3    BLOCKS SENT - BLK FETCH                 0
CONVERSATIONS QUEUED                  0    MESSAGES RECEIVED                   518    MESSAGES IN BUFFER                    949
SQL BOUND FOR REMOTE ACCESS           0    MESSAGES SENT                       528    SWITCH LIMITED BLK PROTOCOL             0
COMMITS RECEIVED                      0    ROLLBACKS RECEIVED                    1
COMMITS SENT                          5    ROLLBACKS SENT                       52
```

Figure 22. Statistics Short Report—DDF

For a complete description of each section, see

- “DRDA REMOTE LOCATION” on page 242
- “DISTRIBUTED ACTIVITY” on page 243

## Statistics Short Report—Page 1

The fields in this section are organized in the following categories:

### HIGHLIGHTS

CREATE THREADS
Number of create thread requests (Q3STCTHD).

This reflects the number of threads created (not including distributed database access threads).

A thread is created at first execution of an SQL statement (assuming that no pre-existing thread for the same authorization ID and plan name exists).

DISTRIBUTED THREADS QUEUED
Number of times a DBAT thread was queued (QDSTQDBT).

This counter indicates that a requester thread was queued because the serving system did not allow enough active remote threads. The tuning parameter is MAXDBAT in DSNZPARM.

**Tuning Tip:** If this situation occurs often, increase MAXDBAT in DSNZPARM on installation panel DSNTIPE; reassemble and restart DB2.

INCREMENTAL BINDS
Number of Incremental BINDs Performed (QXINCRB).

This counter is incremented every time a plan is run that had been bound with the VALIDATE(RUN) option.

**Tuning Tip:** It is generally undesirable to bind a plan with the VALIDATE(RUN) option since all SQL statements must be rechecked for syntax, authority, and access path every time the plan is executed. VALIDATE(RUN) is required if the program is going to CREATE TABLES (for example, work tables) during the execution or if testing is required on a piece of code for which the objects do not yet exist. Otherwise, VALIDATE(RUN) should be avoided, as the cost is nearly that of dynamic SQL.

COMMITS
Total number of commits and rollbacks listed below (Q3STCOMM+Q3STRDON+Q3STSYNC+Q3STABRT).

COMMITS - PHASE 2
Number of successful phase 2 requests (Q3STCOMM).

This is a counter of successful phase 2 commits from IMS or CICS transactions.

This field is not incremented for distributed two-phase commits or single unit of work tasks (for example, TSO, batch).

COMMITS - READ ONLY

Number of read-only commits (Q3STRDON).

This situation occurs only in CICS and IMS when execution of a program has not updated a DB2 resource. When this occurs, DB2 simply increments the read-only counter, performs both phases of the two-phase commit process, and records that the job was read-only. This count does not include CICS synchronous commits.

**Tuning Tip:** This information is useful in determining the read/write ratio of a CICS or IMS system.

SYNC COMMITS

Number of successful single phase commits (Q3STSYNC).

This is a count of all synchronous commits issued by TSO, batch, CAF, and utility programs. CICS applications use both synchronous commits and two-phase commits. IMS uses only two-phase commits.

ROLLBACK

Number of successful rollbacks (Q3STABRT).

This is a count of the number of rollbacks taken by the system to back out a unit of recovery.

Rollbacks occur because of a program abend, application rollback request, deadlock or timeout situation, a -CANCEL THREAD command, or some resource shortage. If it is a resource shortage, a -904 return code message is displayed on the MSTR job log.

## CPU TIMES (SECONDS)

This section shows CPU times for the five DB2 address spaces:

- MSTR
- DBM
- IRLM
- DIST

TOTAL

Total TCB and SRB CPU time for the address space being tracked.

TCB

TCB CPU time (QWSAEJST).

This field provides the CPU time for the address space being tracked.

The information in the CPU time generally represents time which is not allocated to threads and is therefore usually asynchronous overhead.

SRB

SRB CPU time (QWSASRBT).

This field accumulates the CPU time for all SRB tasks assigned to the address space being tracked.

Many DB2 events are scheduled as preemptable SRBs (service request block daughter tasks) for which time can be obtained. They are, therefore, DB2 address space overhead.

## SQL DML

This section shows the number of each of these SQL DML statements:

- SELECT
- INSERT
- UPDATE
- DELETE
- PREPARE
- DESCRIBE
- DESCRIBE TABLE
- OPEN
- FETCH
- CLOSE

The total number of SQL DML statements processed is shown at the bottom of the column.

## SQL DCL

LOCK TABLE

Number of LOCK TABLE statements (QXLOCK).

This counter is incremented every time a LOCK TABLE statement is executed (either SHARE or EXCLUSIVE).

GRANT

Number of GRANT statements (QXGRANT).

This counter is incremented every time a GRANT statement is executed.

**Tuning Tip:** This field is useful for the auditor who wishes to monitor the granting of authority for a specific time period. Other audit traces can be activated to capture which authorities were granted. The catalog can also be queried.

REVOKE

Number of REVOKE statements (QXREVOK).

This counter is incremented every time a REVOKE statement is executed.

**Tuning Tip:** This field is useful for the auditor who wishes to monitor the revocation of authority for a specific time period. Other audit traces can be activated to capture which authorities were revoked.

SET CURRENT SQLID

Number of SET CURRENT SQLID statements (QXSETSQL).

This counter is incremented every time a SET CURRENT SQLID statement is executed.

SET HOST VARIABLE

Number of SET HOST VARIABLE statements (QXSETHV).

This counter is incremented every time a SET HOST VARIABLE statement is executed.

SET CURRENT DEGREE

Number of SET CURRENT DEGREE statements executed (QXSETCDG).

This counter is incremented every time an application program issues a SET CURRENT DEGREE register to enable or disable parallel processing for dynamic SQL.

SET CURRENT RULES

Number of SET CURRENT RULES statements executed (QXSETCRL).

This counter is incremented every time an application program issues a SET CURRENT RULES register to change syntax parsing from SQL rules to ANSI/SQL processing.

**Tuning Tip:** This number may be useful in determining the amount of non-DB2 traffic being executed during this interval period.

SET CURRENT PATH

Number of SQL SET CURRENT PATH statements (QXSETPTH).

(DB2 6.1 and later)

CONNECT TYPE 1

Number of CONNECT TYPE 1 statements executed (QXCON1).

This is a count of the number of CONNECT TYPE 1 statements processed.

Type 1 SQL CONNECT statements allow one updateable site in the connection. The type of Connect statement is specified as a precompiler parameter CONNECT(1) or CONNECT(2).

CONNECT TYPE 2

Number of CONNECT TYPE 2 statements executed (QXCON2).

This is a count of the number of CONNECT TYPE 2 statements processed.

CONNECT TYPE 2 SQL statements allow multiple updateable sites in the distributed connection. The type of Connect statement is specified as a precompiler parameter CONNECT(1) or CONNECT(2).

SET CONNECTION

Number of SET CONNECTION statements executed (QXSETCON).

This counter is incremented every time an application program issues a SET CONNECT SQL verb. This verb is required to perform multi-site updates.

RELEASE

Number of RELEASE statements executed (QXREL).

This counter is incremented every time an SQL RELEASE statement is issued terminating a remote conversation.

**Tuning Tip:** This number is useful in determining the activity of network conversations during the interval period.

CALL

Number of CALL statements executed (QXCALL).

This counter is incremented every time an application program issues a CALL statement to execute a DB2 stored procedure.

ASSOCIATE LOCATOR

Number of ASSOCIATE LOCATOR statements executed (QXALOCL).

These statements get the result set locator value for each result set returned by a stored procedure (DB2 5.1 and later).

HOLD LOCATOR

Number of HOLD LOCATOR statements (QXHOLDL).

(DB2 6.1 and later)

FREE LOCATOR

Number of FREE LOCATOR statements (QXFREEL).

(DB2 6.1 and later)

ALLOCATE CURSOR
Number of ALLOCATE CURSOR statements executed (QXALOCC).

These statements define a cursor and associate it with the result set locator variable (DB2 5.1 and later).

TOTALS
Total number of SQL DCL statements processed.

## SQL DDL

This section shows the number of each of these SQL DDL statements:

- CREATE
- DROP
- ALTER

The total number of SQL DDL statements processed is shown at the bottom of the column.

The number of each of the following statements is also shown in this section below the total of SQL DDL statements:

- COMMENT ON
- RENAME TABLE (DB2 5.1 and later)
- LABEL ON

## RID LIST

RID CURRENT BLOCKS ALLOCATED

Current number of RID blocks in use (QISTRCUR).

This is a snapshot of the number of RID blocks in use.

**Tuning Tip:** If this number is at the RID pool size, check for failures. Increase the DSNZPARM parameter MAXRBLK in DSNZPARM to the maximum possible to support list prefetch and multiple index access paths.

RID HIGH BLOCKS ALLOCATED

Maximum number of RID blocks in use (QISTRHIG).

This value is the maximum number of RID blocks in use at any one time.

**Tuning Tip:** If this value is at the RID pool size, check for failures. Increase the DSNZPARM parameter MAXRBLK in DSNZPARM to the maximum possible to support list prefetch and multiple index access paths.

RID TERM OVER RDS LIMIT

Number of RID pool failures—RID limit (QISTRLLM).

This counter is incremented every time a RID pool failure occurred due to a single set of index entries occupying more than 50% of the RID pool.

**Tuning Tip:** This number should be as close to zero as possible since the query degrades to a table space scan if a failure occurs. Increase the size of the RID pool in DSNZPARM MAXRBLK on installation panel DSNTIPC.

RID TERM TOO MANY CONCURR

Number of RID pool failures—too many concurrent users (QISTRMAX).

This counter is incremented every time a RID pool failure occurred because there were too many concurrent users.

**Tuning Tip:** Increase the DSNZPARM parameter MAXRBLK in DSNZPARM to the maximum possible to support list prefetch and multiple index access paths.

RID TERM - NO STORAGE

Number of RID pool failures—storage exceeded (QISTRSTG).

This counter is incremented every time a RID pool failure occurred due to a lack of storage.

**Tuning Tip:** Increase the DSNZPARM parameter MAXRBLK in DSNZPARM to the maximum possible to support list prefetch and multiple index access paths.

RID TERM - OVER DM LIMIT

Number of RID pool failures—Data Manager limit (QISTRPLM).

This counter is incremented when a very large RID list is encountered. The maximum RID list is 16 million RIDS.

**Tuning Tip:** To avoid this situation, change the SQL statement so a different access path is chosen. If a failure does occur, the query is run as a table space scan.

## LOCKING ACTIVITY

TIMEOUT

Number of lock timeouts (QTXATIM).

This count is incremented every time a DB2 thread waits longer to get a page than the timeout interval. This is specified with DSNZPARM IRLMRWT on installation panel DSNTIPI. By default, it is 60 seconds. Utilities can be allowed to wait several multiples of IRLMRWT.

**Tuning Tip:** Lock timeouts are usually caused by an application failing to commit in time for the other thread to gain access to data on pages. Often the problem can be resolved by reducing the time between commits and putting updateable statements near to their commit logic. Every time a timeout occurs, DB2 writes the holder and suspender to the MSTR job log. Normally, this number should be as close to zero as possible.

DEADLOCK

Number of deadlocks (QTXADEA).

This count is incremented every time DB2 encounters a deadlock situation for which the IRLM must cancel a task involved in a *deadly embrace*.

**Tuning Tip:** Deadlocks are caused by threads requesting access to two resources which can never be resolved. DB2 chooses its victim by selecting the thread that has done the least number of updates. It records the deadlock in the MSTR address space job log. In well-tuned systems, this number should be low.

SUSPEND - LOCK CONFLICT

Number of suspends due to lock conflict (QTXASLOC).

This counter is incremented any time a thread has a conflicting lock request, such as an updater requesting exclusive access to a page being used by another thread.

**Tuning Tip:** In a multitasking system, suspensions occur in the normal course of the events. If applications are well-tuned, taking frequent commits and holding on to resources for the fewest possible instructions, suspensions can be minimized. If suspensions cause frequent timeouts, consider row level locking (DB2 4.1 and later).

SUSPEND OTHER REASONS

Number of suspends due to other conflicts (QTXASOTH).

This number is incremented when DB2 internal processes collide.

**Tuning Tip:** This number is not generally of significance in tuning. Unusually high numbers should be reported to IBM service.

LOCK ESCALATION TO SHARED

Number of lock escalations to shared mode (QTXALES).

This count is incremented every time the number of locks against a single table space exceeds the number set in DSNZPARM NUMLKTS on installation panel DSNTIPJ or in the LOCKMAX clause of the CREATE table space statement (DB2 4.1 and later).

**Tuning Tip:** This is not a normal situation unless using repeatable read (RR). If it occurs often, consider changing the LOCKSIZE or LOCKMAX to a higher value, or consider binding the plan with cursor stability (CS) or uncommitted read (UR).

LOCK ESCALATION TO EXCLUSIVE

Number of lock escalations to exclusive mode (QTXALEX).

This counter is incremented every time the number of updateable locks against a single table space exceeds the DSNZPARM NUMLKTS on installation panel DSNTIPJ or in the LOCKMAX clause of the CREATE table space statement (DB2 4.1 and later). It occurs when the LOCKSIZE parameter is specified as ANY and DB2 has escalated the lock owner to an exclusive lock of the entire table.

**Tuning Tip:** This is an extremely undesirable situation, usually caused by leaving the LOCKSIZE(ANY) default. To resolve this situation, consider making LOCKSIZE(PAGE) or in special situations, LOCKSIZE(ROW) (DB2 4.1 and later). This causes the offending application to take the -904 unavailable resource error rather than cause general unavailability to the rest of the users. This is almost always caused by application failure to commit in a timely fashion and can be resolved by application code changes as well as by DBA action.

DRAIN REQUEST

Number of drain requests (QTXADRNO).

This counter is incremented each time a utility or command requests a serialization against a page set resource.

**Tuning Tip:** This is of significance in determining the amount of utility and command activity that requests some serial access to a resource.

UNSUCCESSFUL DRAIN REQUEST

Number of unsuccessful drain requests (QTXADRUN).

This counter is incremented when a potential drainer (utility or command) cannot obtain use of a page set because the claim count has not dropped to zero within the utility timeout value set in IRLMWRT of DSNZPARM.

This number is significant in determining the number of unsuccessful utility and command processes due to user activity in the resource.

CLAIM REQUEST

Number of claim requests (QTXACLNO).

This number is incremented every time a user executes an SQL statement that increments the use count of a table space, partition, or index space data set.

This number gives an overall level of SQL activity in this system.

UNSUCCESSFUL CLAIM REQUEST

Number of unsuccessful claim requests (QTXACLUN).

This number is incremented every time a user issues a request for a claim to an SQL resource but cannot acquire it, usually because a utility or command DRAIN is on the object being sought.

This number is of some significance in determining contention between SQL and other types of utilities or commands.

LOCK REQUEST

Number of lock requests (QTXALOCK).

This counter is incremented for each call to the IRLM lock manager to acquire a lock on a page or row or to acquire a claim or drain on a data set.

Each lock request is processed by the IRLM. Lock avoidance techniques should show reductions in overall counts and overhead since latches execute totally within DB2.

UNLOCK REQUEST

Number of unlock requests (QTXAUNLK).

This count is incremented when the application has finished processing a page or row, or a claim or drain can be released.

This amount is significant as to cross memory processing and reflects the normal release of resources.

LOCK QUERY REQUEST

Number of query requests (QTXAQRY).

This counter is incremented every time the IRLM gets a request to read data.

**Tuning Tip:** This information is useful in determining the read activity on a system but does not include lock avoidance techniques.

LOCK CHANGE REQUEST

Number of change requests (QTXACHG).

This counter is incremented every time the IRLM is asked to change a lock from one type to another (for example, from ‘S’ to ‘X’).

The information is useful in determining the overall volatility of the system.

LOCK OTHER IRLM REQUEST

Number of other IRLM requests (QTXAIRLM).

This counter is incremented every time the IRLM receives a lock request not included in the other counts.

## EDM POOL

PAGES IN EDM POOL

Number of pages in the EDM pool (QISEPAGE).

This is the size, in 4K pages, of the EDM pool. This is a snapshot value.

This is set in DSNZPARM EDMPOOL on the installation panel DSNTIPC.

FREE PAGES

Number of free pages (QISEFREE).

This counter is a snapshot of how many pages in the EDM pool are not being used by any DBD, SKCT, SKPT, CT, or PT.

**Tuning Tip:** A rough rule of thumb is that in a production system in which the workload is predictable, at least 15% of the EDM pool pages should be free to account for an unanticipated load of a large DBD. This can vary, however, since many shops run more plans than could practically be kept in memory.

EDM POOL FULL FAILURES

Number of failures because EDM pool was full (QISEFAIL).

This counter is incremented when an operation cannot proceed because the EDM pool was full. This is an extremely undesirable situation in which all pages in the Environmental Descriptor Manager pool (EDM pool) are allocated and in use as database descriptors (DBDs), skeleton cursor tables (SKCTs, which are internal copies of plans), skeleton package tables (SKPTs, which are internal copies of packages), and working cursor tables (CTs) and working package tables (PTs). No other operation can be commenced until one or more pages are freed in this pool.

**Tuning Tip:** This number should be as close to zero as possible. If this situation occurs, consider increasing the number of pages in the EDM pool in DSNZPARM, on installation panel DSNTIPC.

CT PAGES USED

Number of pages used for cursor table sections (QISECT).

This is a snapshot of the number of pages allocated to cursor table sections (CTs).

The ideal goal is to keep as many SKCTs, SKPTs, and DBDs in the EDM pool as possible to avoid I/O to the DB2 Directory.

DBD PAGES USED

Number of pages used for DBDs (QISEDBD).

This counter is a snapshot of the number of pages currently occupied by database descriptors (DBDs).

**Tuning Tip:** This count is the most important count of EDM pool objects because of an inherent rule that DBDs must occupy contiguous storage. SKPTs, SKCTs, CTs, or PTs do not have this restriction. This may result in EDM pool fragmentation. Ways of avoiding this problem include keeping DBD sizes to a reasonable size by not allowing more than 100 objects per DB2 database. If the number of DBD pages occupy more than half the EDM pool, consider increasing the size of the pool.

PKG TBL PAGES USED

Number of pages used for package tables (QISEKT).

This is a snapshot of the number of pages allocated to working Package tables (PTs).

This number reflects the amount of storage consumed by working package tables.

SKCT PAGES USED

Number of pages used for skeleton cursor tables (QISESKCT).

This is a snapshot of the number of EDM pool pages allocated to skeleton cursor tables (SKCTs).

Skeleton cursor tables are the DB2 plan information that is read from directory table SCT01.

SKPT PAGES USED

Number of pages used for skeleton package tables (QISESKPT).

This is a snapshot of the number of pages allocated to skeleton package tables (SKPTs).

This value reflects the amount of storage used by skeleton package tables.

CT REQUESTS

Number of requests for cursor table sections (QISECTG).

This number is incremented every time DB2 needs a new 4K section (CT) of the skeleton cursor table (SKCT). The request can be resolved by looking for a copy in the EDM pool; if a copy does not exist, a read operation must be made to the directory table SCT01.

This number reflects how many sections were requested during a specific interval period.

CT NOT IN EDM POOL

Number of times CT sections not found in the EDM pool (QISECTL).

This counter is incremented every time a request for a cursor table section (CT) is made for which a copy in an SKCT does not exist in the EDM pool.

**Tuning Tip:** Every time this occurs, an I/O is scheduled to the SCT01 directory table. It is desirable to reduce the I/O on these very volatile directory page sets.

CT REQUESTS/CT NOT IN POOL

Ratio of the number of requests for cursor table sections to the number of times CT sections not found in the EDM pool (QISECTG/QISECTL).

DBD REQUESTS

Number of requests for DBDs (QISEDBDG).

This number reflects the total number of requests for database descriptors (DBDs).

Requests for DBDs are honored by either finding a working copy previously loaded into the EDM pool or by reading directory table DBD01.

DBD NOT IN EDM POOL

Number of times DBD not found in the EDM pool (QISEDBDL).

This counter reflects the number of times the database descriptor for a table is not found in the EDM pool. It must be read from the catalog table DBD01 into contiguous storage in the EDM pool.

**Tuning Tip:** If possible, the DBDs can be preloaded into the EDM pool by issuing the -DISPLAY DATABASE(*) command to seed the EDM pool right after DB2 starts. This may not account for all fragmentation which can occur because of age, but it can establish a baseline of how large the pool should be sized.

DBD REQ/DBD NOT IN POOL

Ratio of the number of requests for DBDs to the number of times DBD not found in the EDM pool (QISEDBDG/QISEDBDL).

PKG TBL REQUESTS

Number of requests for package table sections (QISEKTG).

This counter is incremented every time a package section (PT) is requested.

**Tuning Tip:** Ideally, an SKPT (skeleton package table) with this PT will already be in the EDM pool. If not, the SKPT must be read from directory table SPT01.

PKG TBL NOT IN EDM POOL

Number of times package table sections not found in EDM pool (QISEKTL).

This counter is incremented every time a request for a skeleton package table section is made on behalf of a calling thread but the SKPT was not found in the EDM pool.

**Tuning Tip:** Ideally, an SKPT (skeleton package table) will already be in the EDM pool. If not, it must be read from directory table SPT01.

PKG TBL REQ/PKG NOT IN POOL

Ratio of the number of requests for package table sections to the number of times package table sections not found in EDM pool (QISEKTG/QISEKTL).

PREP STMT CACHE-INSERTS

Number of inserts into the dynamic statement cache (QISEDSI).

(DB2 6.1 and later)

PREP STMT CACHE-REQUESTS

Number of requests for the dynamic statement cache (QISEDSG).

(DB2 6.1 and later)

PREP STMT CACHE-PAGES USED

Number of pages used for the dynamic statement cache (QISEDSC).

(DB2 6.1 and later)

EDM DATA SPACE-PAGES

Number of pages in the data space used by the EDM pool (QISEDPGE).

(DB2 6.1 and later)

EDM DATA SPACE-FREE PAGES

Number of free pages in the data space free chain (QISEDFRE).

(DB2 6.1 and later)

EDM DATA SPACE-FAILURES

Number of failures because the data space is full (QISEDFAL).

(DB2 6.1 and later)

## LOG ACTIVITY

LOG READS FROM BUFFER

Number of log reads from buffer (QJSTRBUF).

This counter is incremented when DB2 successfully finds the necessary data in a log buffer to perform a backout or recover operation.

This is the best situation if an application abends. The pages necessary to be *undone* are still in memory (the DB2 output log buffer) and backout should take place quickly.

LOG READS FROM ACTIVE

Number of log reads from active logs (QJSTRACT).

This counter is incremented when DB2 successfully finds the necessary data in the active log to perform a backout or recover operation. (The pages are no longer in the log output buffer.)

**Tuning Tip:** IBM’s general rule of thumb is that if storage is not a consideration, plan to have enough active logs on DASD to backout or recover for at least the prior 24 hours, without requiring access to the archive logs. If this is not possible, consider the longest unit of work to be run on the DB2 in question and have enough logs so in the event of an abend, calls to the archive log data set are avoided. It should be noted that backout takes approximately twice the time as the original update since DB2 must write compensation records to the log while reading backwards from it.

LOG READS FROM ARCHIVE

Number of log reads from archive logs (QJSTRARH).

This counter is incremented when a backout or recover operation must go back to the archive logs to accomplish the task.

**Tuning Tip:** This is generally not desirable since tape or cartridge mounts must be performed to recover or back out data from the archive logs. If this number is consistently greater than zero, it is a good idea to increase the number of active logs to avoid unnecessary delays.

READ DELAY/UNAVAIL RESOURCE

Number of read accesses delayed due to resource unavailability (QJSTWUR).

This counter reflects the number of time a recovery or backout was delayed due to tape contention or not enough TCBs to process the number requested.

The -SET ARCHIVE command can increase the number of tape drives available to DB2.

READ ACCESS DELAY-TAPE CONT

Number of log read accesses delayed due to tape contention (QJSTTVC).

This counter is incremented when two or more backout or recover operations call for the same tape volume. DB2 holds and MVS enqueue until the volume is freed.

**Tuning Tip:** Ideally, this number should be zero. Consider using DFHSM or similar media to stage the archive data to DASD if this is a frequent occurrence. DB2 allows multiple tasks to share archive data on DASD as well as look-ahead tape mounts.

ACTIVE LOG WRITE REQUESTS

Number of calls to the log write routine (QJSTBFWR).

This counter is incremented every time log records need to be written.

The number does not represent the number of physical I/Os since DB2 attempts to chain CIs together before writing to the log.

ACTIVE LOG WRITE NOWAIT

Number of nowait log writes (QJSTWRNW).

This counter is incremented when log records are written directly to the log buffers without waiting for the write to the data set.

Unless an application program is seriously taking too many COMMITs, this is the normal asynchronous log processing which does not cost the application any wait time.

BSDS REQUESTS

Number of BSDS access requests (QJSTBSDS).

This counter increments every time the logging subsystem must access the bootstrap data set. DB2 maintains the RBA range for every active and archive log in the system.

**Tuning Tip:** This number will usually be non-zero, since the BSDS must be updated by DB2 every time an archive log process occurs (a copy of the BSDS is REPROed to the first file on the archive medium) and updates are made to the high and low RBA range. Over time, however, look for steadily increasing numbers in this counter. The bootstrap data set is the only key-sequenced VSAM data set (KSDS) used by DB2. As with any other KSDS, CI and CA splits can and do occur. These should be dealt with by REPROing the KSDS to a different file and/or redefining the BSDS as a larger data set.

ACTIVE LOG UNAVAIL BUFFER

Number of waits for unavailable log buffer (QJSTWTB).

This counter is incremented when DB2 must wait for an available output log buffer before it can externalize log records.

**Tuning Tip:** This counter should always be zero, and in all but the busiest DB2 environments will be zero, since it takes a large amount of updated pages to oversaturate the log buffers. Increase the OUTBUFF parameter in DSNZPARM on the DSNTIPL installation screen should this situation occur. All applications wait following a commit until log buffers become available.

Also check that the log write threshold (WRTHRSH) does not exceed the recommended 20% of the total buffers.

ACTIVE LOG CIS CREATED

Number of active log output CIs created (QJSTBFFL).

This counter simply updates the number of 4K CIs created as part of log processing.

**Tuning Tip:** The number is useful in determining the heaviest logging periods in the system.

CONTROL INTERVALS OFFLOADED

Count of control intervals offloaded (QJSTCIOF).

This number reflects the number of active log CIs archived to the archive log medium.

This number will vary per time interval based upon the size of the active log data set. It is generally recommended that an active log be approximately the capacity of a cartridge data set (approximately 300 cylinders of 3390 DASD).

ARCHIVE LOG RD ALLOCATIONS

Number of archive log read allocations (QJSTALR).

This counts every allocation of an archive log for backout or recover operations.

**Tuning Tip:** High numbers indicate that there are not enough active logs properly sized to prevent archive log allocation. In a high performance environment, consider having active logs to hold 24 hours worth of work and archive the logs to media which can be shared in the event of an application failure involving recovery of multiple table spaces. The ideal value is zero.

ARCHIVE LOG WR ALLOCATIONS

Number of archive log write allocations (QJSTALW).

This counts every allocation of an archive log for write.

This is an indication of how many archive logs are written. Larger archive logs mean fewer allocations.

LOOK-AHEAD TAPE MOUNT ATT.

Number of look-ahead tape mounts attempted (QJSTLAMA).

DB2 attempts to pre-stage the second and third archive tape or cartridge volumes with archive data sets to reduce operator mount time. This counter reflects the number of times DB2 had to do this.

**Tuning Tip:** This number should be equal to the number of successful look-ahead tape mounts. Enough tape/cartridge drives must be available to DB2 on the MVS image to make use of this feature.

LOOK-AHEAD TAPE MOUNT SUCC

Number of successful look-ahead tape mounts (QJSTLAMS).

This counter is incremented when DB2 is able to pre-stage the second and third volumes of multiple tapes or cartridges in a backout or recovery operation.

**Tuning Tip:** This number should be equal to the number of look ahead tape mounts attempted. If they are not equal, a shortage of tape drives allocated to DB2 probably exists. To determine the amount of drives available, issue the -DIS ARCHIVE command; if too few drives are allocated in the *count* field, consider issuing the -SET ARCHIVE command to allocate a higher number, assuming the tape and/or cartridge resources are available.

LOG SUSPENDS FOR LOG WRITES

Number of times that a log manager request results in a suspend for a log record that is being written out to the log data sets (QJSTLSUS).

LOG WRITE I/O REQUESTS

Total number of log write I/O requests (Media Manager calls) (QJSTLOGW).

This value includes waits for copy1 and copy2 active log data set writes.

LOG CI WRITTEN

Total number of log CIs written (QJSTCIWR).

This value includes CI rewrites and copy1 and copy2 active log data set writes.

If a given CI is rewritten five times, this counter is incremented by five.

**Tuning Tip:** This counter, multiplied by 4KB and divided by the statistics interval in seconds, represents the number of bytes per second of log data written to the active log data sets. When this value exceeds 1MB/sec per log copy, attention should be paid to log data set I/O tuning.

LOG SERIAL WRITES

Number of serial log write I/O requests (QJSTSERW).

A serial log write I/O request occurs when DB2 rewrites a log CI that was previously written as a partial CI, in a dual logging environment.

This value includes copy1 and copy2 active log data set writes.

**Tuning Tip:** The difference between QJSTSERW and QJSTLOGW represents the number of parallel log write I/O requests. Typically, the first CI in a list of CIs to be written in one start I/O is written serially, and the remaining CIs are written in parallel to both copy1 and copy2 active log data sets. This value is meaningful only when DB2 runs in dual active log mode.

LOG ASYNC WRITES-THRESHOLD

Number of times that an asynchronous log write request was scheduled because the log write threshold was reached (QJSTTHRW).

**Tuning Tip:** It is recommended to use the default write threshold of 20 buffers.

LOG BUFFER PAGEDINS

Number of times that a log output buffer was paged in before it could be initialized (QJSTBPAG).

When a log output buffer is paged in before it is initialized, the log write latch is held.

**Tuning Tip:** A nonzero value could indicate that the log output buffer size is too large or there is insufficient real storage to back the log output buffer size.

# Statistics Short Report—Page 2

The fields in this section are organized in the following categories. For the highlights, see page 194.

## BP TOTALS

GETPAGES

Number of getpage requests (QBSTGET).

This counter is incremented when the Data Manager component of DB2 requests a page (hence *getpage*) from DB2's Buffer Manager. The Buffer Manager first looks into one of its virtual buffer pools, next to hiperpools (if any) and then does a read from DASD.

**Tuning Tip:** The goal in DB2 performance tuning is to keep the ratio of GETPAGEs to real I/O very high. This indicates either good sequential performance or a re-reference of random data, depending on the application.

GETPAGE SEQ REQ

Number of getpage requests for sequential access (QBSTSGT).

This is a count of getpages for all sequential access requesters including sequential prefetch, dynamic sequential prefetch, and list prefetch operations.

**Tuning Tip:** This number is useful in determining what the sequential steal threshold should be for that buffer pool. This number compared with the total number of GETPAGEs indicates the environment for random and sequential processes. Those two numbers can help to set the VPSEQT parameter.

SYNC READ I/O

Number of synchronous read I/Os (QBSTRIO).

This counter is incremented every time the DB2 Buffer Manager could not find a page in global, central, or expanded storage. DB2 must perform a physical read of DASD to obtain the necessary data. The application waits for DB2 to perform the operation.

**Tuning Tip:** Unnecessary read I/Os are one of the principal culprits in poorly tuned applications. While random I/O cannot be avoided, critical re-referenced indexes and tables can be kept in memory. At other times, the strategy is to get the data into memory before the application needs it, typically when processing the data pages sequentially. This is done with prefetch and possibly with parallel processing.

SYNC READ IO SEQ REQ

Number of synchronous I/Os for sequential access (QBSTSIO).

This count is the number of non-sequential pages found while trying to process data sequentially. DB2 must then do random reads.

**Tuning Tip:** This number can indicate data or index fragmentation. It may also indicate buffer pool thrashing. If the pool is too small, sequentially read pages may be stolen before being used. They must then be read in again.

GETPAGES / READ IO

Ratio of the number of getpage requests to the number of synchronous read I/Os (QBSTGET/QBSTRIO).

SEQ PREFETCH REQ

Number of sequential prefetch requests (QBSTSEQ).

This counter is incremented every time a DB2 plan calls for a sequential prefetch operation, which normally attempts to bring in up to 32 pages per read I/O.

Sequential prefetch in this context includes only plans which have indicated a use of sequential prefetch in their EXPLAINs. Dynamic sequential prefetch (where DB2 decides the program is traversing the data sequentially and dynamically brings in the next 32 pages requested) is not included in this count.

SEQ PREFETCH READ IO

Number of asynchronous read I/Os because of sequential prefetch (QBSTPIO).

This counter is incremented for each physical read of data using sequential prefetch for both applications and utilities.

This is not the number of pages read; it simply represents the number of I/O operations caused by sequential processing.

SEQ PREFETCH PAGES READ

Number of asynchronous pages read by sequential prefetch (QBSTSPP).

This counter is incremented with the number of pages read using normal (not dynamic) sequential prefetch.

SEQ PREFETCH PAGES/READ

Ratio of the number of sequential prefetch pages read per sequential prefetch read I/O (QBSTSPP/QBSTPIO).

LIST PREFETCH REQUESTS

Number of list prefetch requests (QBSTLPF).

This number is incremented each time an access path requires that index keys be sorted into data order using the RID pool. List prefetch can be used with non-matching index scans and is always used when multiple indexes are used to access tables. It is also used to sort index data during a hybrid join.

**Tuning Tip:** At BIND time, DB2 computes whether list prefetch should be activated by estimating the number of index RIDS to be scanned and comparing that to the size of the RID pool. If the number exceeds 50%, DB2 does not activate list prefetch. At execution time, if the object would take more than 25% of the RID pool, DB2 disables list prefetch. If this occurs frequently, consider enlarging the RID pool by increasing the MAXRBLK parameter of DSNZPARM on installation panel DSNTIPC.

LIST PREFETCH READ IO

Number of asynchronous read I/Os because of list prefetch. (QBSTLIO).

This is not the number of pages read; it simply represents the number of I/O operations caused by list processing.

LIST PREFETCH PAGES READ

Number of asynchronous pages read by list prefetch (QBSTLPP).

This is a count of the number of index pages read by list prefetch to satisfy non-matching index scans, multiple index access support, and certain types of join to be read into the RID pool for RID pool support.

DB2, upon sorting the RID list created by list prefetch, will then access the data using sequential prefetch to gain the performance boost of processing the data. This is an asynchronous process not charged to the calling application.

LIST PREFETCH PAGES/READ

Ratio of the number of list prefetch pages read per list prefetch read I/O (QBSTLPP/QBSTLIO).

DYNAMIC PREFETCH REQUESTS

Number of dynamic prefetch requests (QBSTDPF).

This field is incremented every time DB2 determines sequential prefetch should be dynamically activated. DB2 analyzes the data accessed to determine whether the last five of eight pages accessed are in sequential order and the application would be benefited by prefetch. DB2 then turns on sequential prefetch until the pages being accessed are no longer in sequential order.

Normally this assists programs by providing read-ahead buffering for processing; GETPAGEs that would have to wait for synchronous I/O now find the page in the buffer pool. This number should be monitored since the more dynamic prefetch requests activated, the more buffer pool resources can be strained. It may be necessary to alter the buffer pool size or sequential steal threshold to handle increased demands of dynamic prefetch.

DYNAMIC PREFETCH READ IO

Number of asynchronous read I/Os because of dynamic prefetch. (QBSTDIO).

This is not the number of pages read; it simply represents the number of I/O operations caused by dynamic processing.

DYNAMIC PREFETCH PAGES READ

Number of asynchronous pages read by dynamic prefetch (QBSTDPP).

This number is the total number of pages accessed asynchronously using sequential prefetch because DB2 dynamically determined by sequential detection that the application was processing the pages sequentially.

**Tuning Tip:** Normally, this aids performance by having read-ahead buffering of pages in the buffer pool. System tuners and DBAs should be aware of when this happens so the number of pages allocated for sequential processing is adequate to support the workload. The VPSIZE (virtual pool size) and the VPSEQT (virtual pool sequential steal thresholds) are the numbers that should be tuned to support the workload.

PREFETCH QTY CUT TO 1/2

Number of times prefetch quantity reduced to 1/2 (QBSTPL1).

This counter is incremented every time DB2 has to reduce the prefetch quantity by 50% because too many parallel tasks have reduced the amount of buffer storage available for sequential processing. For example, instead of getting 32 pages per prefetch, only 16 pages are prefetched.

**Tuning Tip:** Generally, non-zero numbers are undesirable as it indicates there are too many concurrent prefetch processes for parallel processing. This is impacting the effectiveness of prefetch by increasing I/Os. Consider reducing the VPPSEQT or increasing the VPSIZE of the buffer pool.

PAGE UPDATES

Number of page updates (QBSTSWS).

This counter is incremented each time a page is updated in the pool. Once a page is updated, it is *in use* until committed or rolled back. If the same page is updated multiple times, each update intent is counted. Also, this count includes not only updates to data pages but also work file pages, so sort activity increases this count. In a data sharing environment, updated pages may cause buffer invalidation in other member DB2s.

This number reflects update activity. Ideally, DB2 writes updated pages asynchronously at a system checkpoint or through deferred write. The goal is to avoid degrading to synchronous writes.

PG UPDATE / PG WRITTEN

Ratio of the number of page updates to the number of pages written (QBSTSWS/QBSTPWS).

SYNC WRITES

Number of immediate writes (QBSTIMW).

This counter is incremented when a buffer pool reaches 97.5% full of non-stealable buffers. DB2 then attempts to schedule all writes synchronously rather than asynchronously. This time is directly charged to all applications that are updating and seriously degrades performance.

**Tuning Tip:** This number should be zero. If non-zero, consider increasing the size of the buffer pool or ALTERing the heaviest updated page set to a different buffer pool.

ASYNC WRITE IO

Number of asynchronous write I/Os (QBSTWIO).

This number is incremented each time DB2 schedules asynchronous writes through the media manager (VSAM) to DASD. This generally occurs at system checkpoints or deferred write thresholds.

**Tuning Tip:** The number of asynchronous writes should far exceed any synchronous write activity.

PAGES WRITTEN / IO

Ratio of the number of pages written to the number of write I/Os.

HORIZONTAL DEF. WR REACHED

Number of times deferred write threshold reached (QBSTDWT).

This counter is updated when there are more than 50% (DWQT default value) of updated pages in a virtual buffer pool waiting to be externalized. DB2 starts asynchronously writing the updated pages until the number drops below the threshold.

**Tuning Tip:** This can occur during heavy update periods, and by itself is not a sign of serious performance degradation. However, if this number is reached often, it could point to the need to enlarge the virtual pool or hiperpool. The 50% default is modifiable by issuing an -ALTER BUFFERPOOL command for the DWQT threshold.

VERTICAL DEFER WR REACHED

Number of times vertical deferred write threshold reached (QBSTDWV).

This counter is incremented when a single page set takes up more space than the user-defined vertical deferred write threshold (by default 10%). The goal is to prevent any single data set assigned to a buffer pool from holding on to updated buffers at the expense of the other page sets occupying the pool. Each time this occurs, DB2 writes these buffers to DASD with chained write I/O.

**Tuning Tip:** The DBA and system tuner need to know the characteristics of the buffer pool and the page sets assigned to it. If a single page set is assigned to a buffer pool, the vertical deferred write threshold can be raised. The goal is to prevent unnecessary DASD I/O caused by a too-low threshold. In general, if this threshold is reached often, the virtual buffer pool size or vertical deferred write threshold should be increased.

DM CRITICAL REACHED

Number of times Data Manager Buffer Critical reached (QBSTDMC).

This counter is incremented when a buffer pool contains 95% or more of non-stealable pages. Sequential prefetch was turned off at 90%. At 95%, DB2 parses rows instead of 4K pages. This becomes evident because more than one GETPAGE can be issued for the same page.

**Tuning Tip:** This is a sign of serious performance stress on the buffer pool. The choice of fixes are to increase the virtual pool size or to allow fewer updates during a specified time period. At 95%, the CPU overhead to read and write rows becomes extreme and is very noticeable to users.

SUCCESSFUL OPEN

Number of data sets physically opened (QBSTDSO).

This counter is incremented each time a page set is physically opened.

MVS OPEN processing of a VSAM data set can be extremely expensive. The goal is to keep frequently used page sets open. While every page set must be opened at least once, the goal is to keep the heavily used page sets open as long as possible. DB2 uses a pseudo-close technique to avoid physical closes.

DB2 reduces the possible impact on recovery of large numbers of open data sets by internally shifting the R/W nature of the page set to read-only if no activity has taken place in the last five checkpoints or 60 minutes.

BPOOL HIT RATIO-ALL(%)

Buffer pool hit ratio with prefetch.

This value is calculated as

(Total Pages - Total I/O) / Total Pages x 100.0

where

Total Pages is the number of getpage requests (QBSTGET).

Total I/O is the sum of

- Synchronous read I/Os (QBSTRIO)
- Asynchronous pages read by sequential prefetch (QBSTSPP)
- Asynchronous pages read by list prefetch (QBSTLPP)
- Asynchronous pages read by dynamic prefetch (QBSTDPP)

BPOOL HIT RATIO-RANDOM(%)

Random buffer pool hit ratio.

This value is calculated as

(Sync Pages - Sync I/O) / Sync Pages x 100.0

where

Sync Pages is the number of synchronous getpage requests (QBSTGET - (QBSTSGT + QBSTNGT)).

Sync I/O is the number of synchronous read I/Os (QBSTRIO - QBSTSIO).

## SYSTEM SERVICES

IDENTIFY REQUEST

Successful identify requests (Q3STIDEN).

This reports the number of threads that have gone through successful connection processing to the DB2 being observed. Connections can be from any supported environment, such as TSO, IMS, CICS, CAF (call attach), or a utility.

**Tuning Tip:** Watch for peak time spike periods. The DB2 DSNZPARM parameters CTHREAD, IDFORE, IDBACK, and other distributed parameters should be adjusted so the system can respond to the maximum load expected.

CREATE THREAD

Create thread requests (Q3STCTHD).

This reflects the number of threads created (not including distributed database access threads).

A thread is created at first execution of an SQL statement (assuming that no pre-existing thread for the same authorization ID and plan name exists).

CREATE THREAD THAT WAITED

Number of create thread requests queued (Q3STCTHW).

This field is a counter of how many times the maximum thread count (CTHREAD in DSNZPARM) was reached and a user had to wait to acquire an available thread. It does not include DBATs.

**Tuning Tip:** The general rule is to keep CTHREAD high enough to avoid thread queuing. However, if memory resources are tight (and the users are willing to live with the pain), you can limit CTHREAD to an arbitrary number. This controls memory allocation for thread creation at the expense of user wait time and some overhead in keeping track of the queued threads for scheduling.

SIGNON

Successful signon events (Q3STSIGN).

This field has meaning only when applied to CICS and IMS. It represents the number of times an existing thread has gone through new user signon reusing an existing thread.

**Tuning Tip:** The cost of thread creation can be quite considerable in CICS transactions and IMS WFI BMP transactions. In general, it is desirable to reuse an existing thread rather than suffer the destruction and recreation of a new thread. High numbers generally indicate good thread reuse. Tuning factors which can impact this are GRANTing plans to PUBLIC (avoiding an authorization check) or utilizing the authorization cache by specifying a large enough CACHESIZE at bind time to keep as many user IDs in the EDM pool as possible.

TERMINATE

Successful terminate thread requests (Q3STTERM).

This number reflects thread termination as a result of a program releasing resources or a thread dropping its access level from *thread active* back to signon and back to identify status. This count will be higher than the create thread count.

In online transactions, keep an eye on thread creations and terminations for similar transactions. Thread reuse may be achieved in CICS by raising the THRDS count in the CICS resource control table; in an IMS environment, a Wait-for-Input WFI) BMP can be a useful method of avoiding costly thread creations and terminations.

ROLLBACK

Successful rollbacks (Q3STABRT).

This is a count of the number of rollbacks taken by the system to back out a unit of recovery.

Rollbacks occur because of a program abend, application rollback request, deadlock or timeout situation, a -CANCEL THREAD command, or by some resource shortage. If it is a resource shortage, a -904 return code message is displayed on the MSTR job log.

PREPARE TO COMMIT (PHASE 1)

Successful prepare to commits (Q3STPREP).

This is a count of the number of *prepare to commit* requests for a two-phase commit unit of work, which includes CICS update and IMS transactions. The prepare to commit is the result of the end of phase one which causes log records to be externalized. This counter is appropriate only for two-phase commit operations.

The use of this field is to determine how much forced logging is occurring as a result of end of phase 1 processing. The number of prepares for phase 1 minus the successful commit phase 2 requests is the indoubt count. However, since some phase 2 operations do not complete in the same time interval, the number of prepares for phase 1 do not always agree with successful phase 2 requests.

COMMIT (PHASE 2)

Successful phase 2 requests (Q3STCOMM).

This is a counter of successful phase 2 commits from IMS or CICS transactions.

This field is not incremented for distributed two-phase commits or single unit of work tasks (for example, TSO, batch).

READ-ONLY COMMIT

Number of read-only commits (Q3STRDON).

This situation occurs only in CICS and IMS when execution of a program has not updated a DB2 resource. When this occurs, DB2 simply increments the read-only counter, performs both phases of the two-phase commit process, and records that the job was read-only. This count does not include CICS synchronous commits.

**Tuning Tip:** This information is useful in determining the read/write ratio of a CICS or IMS system.

SYNCH COMMIT (SINGLE PHASE)

Successful single-phase commits (Q3STSYNC).

This is a count of all synchronous commits issued by TSO, batch, CAF, and utility programs. CICS applications use both synchronous commits and two-phase commits. IMS uses only two-phase commits.

INDOUBT

Total number of indoubt units of recovery (Q3STINDT).

This is a count of indoubt threads, caused when a failure occurs after a successful prepare but before a successful commit. The failure can occur in the address space of the application, the transaction manager, DB2, or a distributed requester / server.

Most indoubt situations are resolved automatically when the participants are restarted. These are counted in *successful indoubt resolutions* (Q3STRIUR). However, in some cases, such as an operator cold starting one participant, manual resolution may be required.

INDOUBT RESOLVED

Successful indoubt resolutions (Q3STRIUR).

This is a count of successful indoubt thread resolutions. Indoubt situations arise when a failure occurs after a successful prepare but before a successful commit. The status of retained locks against resources cannot be resolved until the coordinator and all participants have been recovered/restarted. This resolution usually occurs automatically, as reflected in this count.

This field applies only to CICS and IMS transactions participating in two-phase commit or in distributed processing using DRDA level 2 processing. In some cases, such as an operator cold starting one participant, manual resolution may be required. Another counter shows the total number of indoubt threads (Q3STINDT).

END OF MEMORY

Abnormal allied memory end of memory (Q3STMEOM).

This counter reflects program abends due to end of memory situations (such as an MVS FORCE command or lack of region to successfully abend).

Non-zero numbers in this counter should be investigated to determine why FORCE commands were issued or an X78 abend occurred.

END OF TASK

Abnormal allied memory end of task (Q3STMEOT).

This field is a counter of those tasks which have abended while connected to DB2. The counts reflected should equal the number of Abnormal EOT messages in the MSTR job log.

**Tuning Tip:** This field is useful in determining how many abends occurred.

DB2 CHECKPOINT

Number of DB2 checkpoints (QWSDCKPT).

This counter describes the number of DB2 system checkpoints taken.

During checkpoint processing, important events are written to the log, including internal status information. This information is needed for recovery processing if DB2 abends or MVS terminates without proper shutdown.

## GLOBAL LOCKING ACTIVITY

This section is available for DB2 4.1 and later only.

PLOCK REQUESTS

-LOCK

Number of lock requests for P-locks (QTGSLPLK).

This counter is incremented when a physical lock (P-lock) is acquired in the DB2 data sharing environment at the data set level to allow DB2 to determine whether an object is of inter-DB2 interest. Concurrency is still managed as before with transaction (or logical) locks (L-locks).

A P-lock is acquired for the first transaction that accesses an object (table space, index space, partition) but it is owned by the subsystem. Page P-locks are used for row-level locking. Inter-DB2 interest occurs only when at least one DB2 has update interest and one or more others have read interest. P-locks never cause timeouts, as the lock mode can be negotiated.

-CHANGE

Number of change requests for P-locks (QTGSCPLK).

This counter is incremented when a P-lock has been established and the IRLM associated with this DB2 determines that the nature of that lock must be changed (for example from share to exclusive) because the usage of the page set in that DB2 has changed.

The number of P-lock changes reflects the changing status of page sets as access modes change and the lock state is negotiated between data sharing members.

-UNLOCK

Number of unlock requests for P-locks (QTGSUPLK).

This number represents the number of IRLM unlock requests because a page set is no longer in use.

This number represents the number of times a page set has been unlocked in a data sharing environment.

XES SYNC REQ

-LOCK

Number of XES synchronous lock requests (QTGSLSLM).

This number is incremented for both L-locks and P-locks when a request for a lock on an object of inter-DB2 interest is sent to MVS Cross-System Services. This occurs synchronously under the user's execution unit.

This number reflects the request for locks of inter-DB2 interest after it is determined that no intrasystem locks exist. This count is not incremented if a suspension occurs.

-CHANGE

Number of XES synchronous change requests (QTGSCSLM).

This number is incremented for both L-locks and P-locks when a change request for a lock on an object of DB2 intersystem interest is sent to MVS Cross-System Services.

This number reflects the number of change requests for locks of inter-DB2 interest. This count is not incremented if a suspension occurs.

-UNLOCK

Number of XES synchronous unlock requests (QTGSUSLM).

This number is incremented for both L-locks and P-locks when an unlock request to an object of inter-DB2 interest is sent to MVS Cross-System Services. It is incremented for each resource for which the global lock is released, rather than once for a generic unlock request as is done for local locks.

This number reflects all locks which are released through MVS Cross-System Services. This count is not incremented if a suspension occurs.

SUSPENDS

-IRLM

Number of global suspends - IRLM (QTGSIGLO).

This number is incremented when IRLM global lock states were in conflict on the same resource (page set, page, row), where the lock was held by another DB2 data sharing group member.

This counter is a measure of inter-DB2 contention. All locks of intrasystem contention are resolved prior to the IRLM checking global contention. This is the number of true waits for another thread on a different MVS that has an incompatible lock type to the requester. Global contention requires intersystem communication to resolve the conflict. Data sharing is designed to optimize the quick granting of locks where no contention exists. Try to keep total contentions (IRLM, XES, and false) to less than 2% of the total lock requests.

-XES

Number of global suspends - XES (QTGSSGLO).

This counter is a measure of MVS XES global resource contention.

MVS XES lock states were in conflict but IRLM lock states were not. XES is aware only of exclusive or share locks, where IRLM has many lock states.

-FALSE

Number of global suspends due to false contention (QTGSFLSE).

This counter is incremented every time XES detects contention on the resource hash class but there was no real contention on the resource itself.

**Tuning Tip:** MVS Cross-System Services uses a hash lock table in the coupling facility to provide efficient determination of whether a resource is locked. This hash table points to a number of synonyms (different resource names can hash to the same entry). Contention at the hash entry is considered false contention. The more resources declared to be of inter-DB2 interest, the more chances there are of this condition occurring. If false contentions are more than half of the total global contentions (suspensions for IRLM, XES and false), the CF lock structure size may be too small for the workload.

GLOBAL LOCK CONTENTION(%)

Percentage of global lock contentions.

This value is calculated as

Global Suspends / Total Global Locks x 100.0

where

Global Suspends is the total number of global lock contentions (QTGSIGLO + QTGSSGLO + QTGSFLSE).

Total Global Locks is the sum of

- Lock requests (QTGSLSLM)
- Change requests (QTGSCSLM)
- Unlock requests (QTGSUSLM)
- Global lock contentions (QTGSIGLO + QTGSSGLO + QTGSFLSE)

FALSE CONTENTION(%)

Percentage of global lock false contentions.

This value is calculated as

False Contentions / Global Suspends x 100.0

where

False Contentions is the total number of global lock suspensions due to false contention (QTGSFLSE).

Global Suspends is the total number of global lock contentions (QTGSIGLO + QTGSSGLO + QTGSFLSE).

INCOMPATIBLE RETAINED LOCK

Number of global requests denied due to incompatible retained lock (QTGSDRTA).

This number is incremented every time global lock services denies a lock request because an incompatible lock type has been retained on the requested resource.

This number reflects the instances in which access to a page or row cannot be obtained because another thread on another system had access to the resource requested but the other system failed. The CF retains locks until the other system comes up to complete the unit of work.

NOTIFY MESSAGES -SENT

Number of notify messages sent (QTGSNTFY).

This number represents the number of messages passed to the coupling facility so that, for example, other members can be notified that DBDs have been changed due to CREATE, ALTER, or DROP.

This is the outbound traffic to the coupling facility.

NOTIFY MESSAGES -RECEIVED

Number of notify messages received (QTGSNTFR).

This number represents the number of messages received from the coupling facility.

This is the inbound traffic from the coupling facility.

XES ASYNC REQ -RESOURCES

Number of resources propagated asynchronously to XES (QTGSKIDS).

This counter is incremented whenever a DB2 data sharing P-lock or L-lock (page set lock or row/page lock) is propagated by the IRLM to MVS Cross-System Services asynchronously.

This situation occurs when new inter-DB2 interest occurs on a parent resource (such as a page set) or when a request completes after the requester’s execution unit has already been suspended.

NEGOTIATE

-PGSET/PARTITION

Number of P/P P-lock negotiations (QTGSPPPE).

This counter is incremented when the participating DB2 has to change the lock type of a page set / partition P-lock because another thread on another system changed its intent on this object from SHARE to EXCLUSIVE or from EXCLUSIVE to SHARE.

**Tuning Tip:** Each DB2 in a data sharing group may need to change P-locks as part of a negotiation process so that physical locks never cause timeouts or deadlocks.

-PAGE PLOCK

Number of Page P-lock negotiations (QTGSPGPE).

This counter is incremented when the participating DB2 has to change a page lock type because another thread on another system has changed its intent on this object from SHARE to EXCLUSIVE or EXLUSIVE to SHARE. Page P-locks are used for row-level locking.

**Tuning Tip:** Each DB2 in a data sharing group may need to change P-locks as part of a negotiation process so that physical locks never cause timeouts or deadlocks.

-OTHER LOCK

Number of other P-lock negotiations (QTGSOTPE).

This counter is incremented when anything other than a page set or page lock is negotiated between participating DB2 subsystems.

These events represent other contention which may exist other than page set or page P-locks.

-PLOCK CHANGE REQ

Number of P-lock change requests during P-lock negotiation (QTGSCHNP).

This counter is incremented when a participating DB2 must communicate a lock change request to other DB2s as a result of negotiation for access to an object.

This is a count of the number of P-Lock changes issued during P-Lock intersystem negotiation.

PLOCK/NOTIFY

-MAX ENGINES
Maximum number of engines available for lock/notify exit requests (QTGSPEMX).

This number reflects the number of tasks allocated to tracking intersystem lock negotiation.

This is a count of the facilities available to deal with lock contention from the global DB2 data sharing perspective.

-NO ENGINE
Number of times no engine available for lock/notify exit requests (QTGSPEQW).

This number is incremented when the number of requests for lock or notify exit processing exceeds the maximum number of engines available.

**Tuning Tip:** This number indicates that there are too many intersystem locking and notification requirements based on the number of resources available for global lock management. It may be appropriate to reduce the number of resources being shared.

## QUERY PARALLELISM

### MAX DEGREE OF PARALLELISM

Maximum degree of parallel processing executed (QXMAXDEG).

This counter is set to the high-water mark among all parallel groups executed with I/O or CP parallelism.

Degree measures the number of parallel processes (I/O streams in DB2 3.1 / 4.1, CP tasks in DB2 4.1) active for a parallel group.

### PARALLEL GROUPS EXECUTED

Number of parallel groups executed (QXTOTGRP).

This counter reflects the total number of parallel groups executed.

**Tuning Tip:** This number can provide the tuner with a good idea when peak periods exist which might require more buffer pool size or a higher parallel processing percentage of a buffer pool might be necessary.

### PLANNED DEGREE

Number of parallel groups executed at planned degree (QXNORGRP).

This counter is incremented when the number of parallel tasks (degree) at execution time equal the number of parallel tasks planned at BIND time.

**Tuning Tip:** The higher this number is, the better DB2 is tuned. This is the ideal situation.

### REDUCED DEGREE - NO BUFFER

Number of parallel groups degraded due to buffer shortage (QXREDGRP).

This counter is incremented when the buffer pool does not have enough buffer storage to support as many degrees of parallel processing as had been planned. DB2 checks buffer allocations at both BIND and execution time. It assumes there will be buffers set aside for parallel processing. If at execution time, a similar number of buffers do not exist, DB2 degrades the parallel processes to a lesser degree or to no parallelism.

**Tuning Tip:** Three parameters can be altered (ALTER BUFFERPOOL command) to resolve this situation. The overall size of the buffer pool is controlled by the VPSIZE (virtual pool size) parameter. The amount of sequential buffers is set by the VPSEQT (virtual pool sequential threshold) parameter. In the amount of buffers reserved for sequential processing, a reserve of buffers available for parallel processing must be maintained by the VPPSEQT (virtual pool parallel sequential threshold) parameter. If a significant number of parallel processes are degraded due to buffer shortage or contention, consider using a different buffer pool or altering the buffer pool used so that sufficient parallel sequential buffers are present.

FALLBACK - NO BUFFER

Parallel group fallback to sequential due to buffer shortage (QXDEGBUF).

This counter is incremented when the buffer pool does not have enough buffers to support parallel processing. DB2 checks buffer allocations at both BIND and execution time. It assumes there will be buffers set aside for parallel processing. If at execution time, a similar number of buffers do not exist, DB2 degrades the parallel processes to a lesser degree or no parallelism.

**Tuning Tip:** Three parameters can be altered (ALTER BUFFERPOOL command) to resolve this situation. The overall size of the buffer pool is controlled by the VPSIZE (virtual pool size) parameter. The amount of sequential buffers is set by the VPSEQT (virtual pool sequential threshold) parameter. In the amount of buffers reserved for sequential processing, a reserve of buffers available for parallel processing must be maintained by the VPPSEQT (virtual pool parallel sequential threshold) parameter. If a significant number of parallel processes are degraded due to buffer shortage or contention, consider using a different buffer pool or altering the buffer pool used so that sufficient parallel sequential buffers are present.

FALLBACK - NO ESA

Parallel group fallback to sequential due to lack of sort assist (QXDEGESA).

This counter is incremented when DB2 detects that the hardware sort assist facility is not present to logically partition the DB2 temporary DSNDB07 work files. The parallel sort operation falls back to sequential.

**Tuning Tip:** This situation will occur until the necessary hardware assist is purchased.

FALLBACK - CURSOR

Parallel group fallback to sequential due to updateable cursor (QXDEGCUR).

This counter is incremented when DB2 detects a cursor that is not clearly read-only and falls back from the planned parallel processing to sequential access.

**Tuning Tip:** To resolve this problem, the application program should have a cursor which is unambiguously read-only, with updates done through another cursor or statement.

FALLBACK - ENCLAVE

Parallel group fallback to sequential due to enclave services unavailable (QXDEGENC).

This counter is incremented when DB2 detects that MVS 5.2 enclave support is unavailable to support parallel CP processing. The parallel group falls back to sequential.

MVS enclave support sets objectives for parallel tasks to perform within service goals set by management in the MVS Workload Manager. The DB2 parallel tasks run as enclave SRBs. The solution is to migrate this system to MVS 5.2 as soon as practical to do so.

PRLL GROUPS INTENDED

Number of parallel groups that DB2 intended to run across the data sharing group (QXXCBPNX).

This count is incremented on the parallelism coordinator only at run time (DB2 5.1 and later).

SINGLE DB2-COORDINATOR=NO

Number of parallel groups executed on a single DB2 due to one of the following reasons (QXCOORNO):

- When the plan or package was bound, the COORDINATOR subsystem parameter was set to YES, but the parameter is set to NO when the program runs.
- The plan or package was bound on a DB2 with the COORDINATOR subsystem parameter set to YES, but the program is being run on a different DB2 that has the COORDINATOR value set to NO.

(DB2 5.1 and later)

SINGLE DB2-ISOLATION=RR

Number of parallel groups executed on a single DB2 because the plan or package was bound with an isolation value of repeatable read or read stability (QXISORR) (DB2 5.1 and later).

DB2 MEMBER SKIPPED-NO BUF

Number of times that the parallelism coordinator had to bypass a DB2 when distributing tasks because there was not enough buffer pool storage on one or more DB2 members (QXXCSKIP).

This field is incremented only on the parallelism coordinator, and it is incremented only once per parallel group, even though it is possible that more than one DB2 has a buffer pool shortage for that parallel group (DB2 5.1 and later).

**Tuning Tip:** The purpose of this count is to indicate that there are not enough buffers on one or more members. Therefore, this count is incremented only when the buffer pool is defined to allow parallelism. For example, if VPXPSEQT=0 on an assistant, DB2 does not send parallel work there, but this count is not incremented.

REFORMULATED GROUPS-CONFIG

Total number of parallel groups for which DB2 reformulated the parallel portion of the access path because the sysplex configuration at run time was different from the sysplex configuration at bind time (QXREPOP1).

This counter is incremented only by the parallelism coordinator at run time.

(DB2 6.1 and later)

REFORMULATED GROUPS-NO BUFFER

Total number of parallel groups for which DB2 reformulated the parallel portion of the access path because there was not enough buffer pool resource (QXREPOP2).

This counter is incremented only by the parallelism coordinator at run time.

(DB2 6.1 and later)

## OPEN/CLOSE ACTIVITY

DATASETS CURRENTLY OPEN

Number of data sets currently open (QTDSOPN).

This snapshot count reflects all open table space and index space data sets at the time the statistics record was cut.

DB2 DSNZPARM DSMAX specifies the total storage which can be allocated for open data sets. This number cannot exceed 10,000, unless the requisite OS/390 support is available.

MAX D.S. OPEN CONCURRENTLY

Maximum number of open data sets (QTMAXDS).

This number reflects the high-water mark of all open data sets.

When DB2 reaches 99% of the DSMAX value, it starts physically closing page sets, first those with CLOSE(YES), then those with CLOSE(NO), using a least recently used (LRU) algorithm. This value can be greater than DSMAX.

REOPEN FROM SLOW CLOSE

Number of times a deferred close data set is reused (QTREOPN).

This counter is incremented each time a data set is accessed that was on the deferred close queue. Although the data set was not in use, it was not physically closed, so it did not have to be reopened.

**Tuning Tip:** The higher this number is, the better the system use of deferred close.

## GLOBAL BP TOTALS

This section is available for DB2 4.1 and later only.

SYNC RD INV BUFF

Number of synchronous coupling facility reads for *invalid buffer* in each of these circumstances:

- WITH DATA

A data sharing coupling facility synchronous read was issued because a local virtual buffer pool or hiperpool had a page marked as invalid. The page existed in the group buffer pool and was returned (QBGLXD).

This occurs normally in a data sharing complex in which data must be shared. The correct data is refreshed from the group buffer pool.

- NO DATA

A data sharing coupling facility synchronous read was issued because a local virtual buffer pool or hiperpool had a page marked as invalid but no data was returned (QBGLXR + QBGLXN (DB2 Version 4 only)).

SYNC RD NOT FOUND

Number of synchronous coupling facility reads for *buffer not found* in each of these circumstances:

- WITH DATA

A data sharing coupling facility synchronous read was issued because a page was not found in the local virtual buffer pool or hiperpool but the page existed in the group buffer pool and was returned (QBGLMD).

This occurs normally in a data sharing complex in which data must be shared. The correct data is refreshed from the group buffer pool.

- NO DATA

A data sharing coupling facility synchronous read was issued because a page could not be found in a local virtual buffer pool or hiperpool and no data was returned (QBGLMR + QBGLMN (DB2 Version 4 only)).

ASYNC READ

Number of asynchronous coupling facility reads in each of these circumstances:

- DATA RETURNED

A page existed in a global buffer pool and was returned (QBGLAD).

This is on behalf of prefetch. It occurs normally in a data sharing complex in which data must be shared. The correct data is refreshed from the group buffer pool.

- NO DATA

A data sharing coupling facility asynchronous read was issued but no data was returned from the group buffer pool (QBGLAR + QBGLAN (DB2 Version 4 only)).

This is on behalf of prefetch.

SYNC PAGES WRITTEN

Number of synchronous pages written to the group buffer pool in each of these circumstances:

-CHANGED

A changed page was written to the group buffer pool (QBGLSW).

Updated pages must be written to the coupling facility when the object is of intersystem interest so other DB2s can refresh their invalidated buffers. Only changed pages are written when GBPCACHE is set to the default of CHANGE.

-CLEAN

A clean page was written to the group buffer pool (QBGLWC).

This is done only when GBPCACHE is set to ALL.

**Tuning Tip:** While this can be expensive, it does post the group buffer pool with clean pages and may be useful for small tables with high intersystem read interest to reduce contention.

ASYNC PAGES WRITTEN

Number of asynchronous pages written to the group buffer pool in each of these circumstances:

-CHANGED

A local changed page was written asynchronously to the group buffer pool (QBGLAW).

This is initiated by normal update processing in a data sharing complex.

-CLEAN

A clean page was written asynchronously to the group buffer pool (QBGLAC).

This is done for page sets and partitions defined with GBPCACHE ALL. All clean pages can be kept in the coupling facility to improve access to data sets with high contention.

CASTOUT

-PAGES WRITTEN

Number of pages cast out from the GBP (QBGLRC).

A data sharing complex periodically externalizes updated pages to DASD in a process known as castout.

The castout data is externalized for various reasons:

- A GBP castout threshold is reached (similar to a virtual buffer pool reaching its deferred write threshold)
- A class castout threshold is reached (similar to a data set reaching its vertical deferred write threshold)
- A GBP checkpoint is triggered
- There is no more inter-DB2 interest in the page set

-NO ENGINE

Number of times a castout engine was unavailable (QBGLCN).

This counter is incremented when the coupling facility could not successfully cast out its pages because enough castout engines were not available.

Workloads need to be shifted to make sure the coupling facility can successfully externalize its data. DB2 casts out the pages when an engine becomes available.

-CLASS THRESHOLD

Number of times GBP castout was initiated because the class castout threshold was reached (QBGLCT).

Data Sharing GBP castout processing is initiated because the number of changed pages for a castout class queue exceeded the class threshold. This causes updated pages to be read by the *owning* DB2 from the GBP and to be written to DASD.

**Tuning Tip:** This process is similar to vertical deferred write processing on a local DB2 and is a normal way to externalize updated pages. The default class castout threshold is a single class (one or more page sets / partitions) occupying 10% or more of the group buffer pool. This threshold is tunable.

-GBP THRESHOLD

Number of times GBP castout was initiated because the GBP castout threshold was reached (QBGLGT).

Data Sharing GBP castout processing is initiated because the number of changed pages in the GBP exceeded the GBP castout threshold. This causes updated pages to be read by the *owning* DB2 from the GBP and to be written to DASD.

**Tuning Tip:** This process is similar to deferred write processing on a local DB2 and is a normal way to externalize updated pages. The default GBP castout threshold is 50%. This threshold is tunable.

NO WRITE ENGINE

Number of times a coupling facility write engine was unavailable (QBGLSU).

This counter is incremented when the coupling facility exceeds the number of write engines required.

**Tuning Tip:** This number should be extremely low. It may be necessary to reduce the number of page sets being shared.

READ FAIL -STORAGE

Number of failed coupling facility reads due to lack of storage (QBGLRF).

This counter is incremented when there is not enough storage allocated to the coupling facility to manage that required by the group buffer pools.

**Tuning Tip:** This indicates that the group buffer pools are overcommitted. Reduce the size of group buffer pools unless more coupling facility memory can be acquired to support the group buffer pools.

WRITE FAIL -STORAGE

Number of failed coupling facility writes due to lack of storage (QBGLWF).

This counter is incremented when there is not enough storage allocated to the coupling facility to complete all write requests.

**Tuning Tip:** The group buffer pools are overcommitted. Reduce the size of the group buffer pools unless more coupling facility memory can be acquired to support them.

RD STG STATS(5.1)/OTHER(4.1)

Number of other coupling facility interactions experienced during castout processing (QBGLOS).

GBP CHECKPOINT

Number of group buffer pool checkpoints triggered by this member (QBGLCK) (DB2 5.1 and later).

GBP REBUILD

Number of group buffer pool rebuilds in which this member participated (QBGLRB) (DB2 5.1 and later).

UNLOCK CASTOUT

Number of coupling facility requests to unlock the castout lock for castout I/Os that have completed (QBGLUN).

As pages are in the process of being cast out to DASD, they are *locked for castout* in the coupling facility. The castout lock is not an IRLM lock; its purpose is to enforce that only one system is doing castout for a given page at a time (DB2 5.1 and later).

**Tuning Tip:** DB2 usually includes multiple pages in a write I/O request for castout. Therefore, this value should always be less than or equal to the number of pages cast out. It will be significantly less if multiple pages are written per I/O, so this ratio gives a good indication of how castout is performing.

READ CASTOUT CLASS

Number of coupling facility requests to read the castout class (QBGLCC).

DB2 uses this number to determine which pages that belong to a given page set or partition are cached as *changed* in the group buffer pool and need to be cast out (DB2 5.1 and later).

This request is issued by the page set of partition castout owner. It is also issued by the group buffer pool structure owner when the GBPOOLT threshold is reached.

READ CASTOUT STATISTICS

Number of coupling facility requests to read the castout statistics (QBGLCS).

The group buffer pool structure owner issues this request when the group buffer pool threshold GBPOOLT is reached to determine which castout classes have changed pages. Generally, this occurs only once or twice for each occurrence of the GBPOOLT threshold being reached (DB2 5.1 and later).

DELETE DIR/DATA ENTRIES

Number of group buffer pool requests to delete all directory and data entries for a page set or partition (QBGLDN).

DB2 issues this request when it converts a page set or partition from GBP-dependent to non-GBP-dependent. For objects defined with GBPCACHE ALL, DB2 issues this request when the first DB2 member opens the object (DB2 5.1 and later).

READ DIRECTORY INFORMATION

Number of coupling facility requests to read directory information (QBGLRD) (DB2 5.1 and later).

The group buffer pool structure owner issues this request for GBP checkpoints to read the directory entries of all changed pages in the group buffer pool so that the oldest recovery LRSN can be recorded and used for recovery purposes in case the group buffer pool should fail. (This recovery LRSN is displayed in message DSNB798I.)

**Tuning Tip:** This request may need to be issued several times for each GBP checkpoint in order to read the directory entries for all changed pages. If you see an abnormally high number here, it might be that these requests are being cut short by the coupling facility model-dependent timeout criteria. To help alleviate this, you should consider upgrading those coupling facilities to CFLEVEL 2 or above.

REGISTER PAGE

Number of coupling facility requests to register interest to the GBP for a single page (QBGLRG).

These are *register-only* requests. DB2 is not requesting that any data be returned for the page because it knows that the page is not cached in the group buffer pool (DB2 5.1 and later).

The *register page* request is made only to create a directory entry for the page for cross-invalidation when downgrading the P-lock on a page set or partition from S mode to IS mode, or from SIX mode to IX mode.

UNREGISTER PAGE

Number of coupling facility requests to unregister interest to the GBP for a single page (QBGLDG).

This is generally done as DB2 steals pages from the local buffer pool that belong to GBP-dependent page sets or partitions (DB2 5.1 and later).

REGISTER PAGE LIST

Number of requests to register a page list in the coupling facility (QBGLAX).

DB2 prefetch can do this only if the member is running on MVS 5.2 or later and if the group buffer pool is allocated in a coupling facility with CFLEVEL=2 or above (DB2 4.1 and later).

REGISTER PAGE LIST-RD CHNGE

Number of coupling facility reads to retrieve a changed page for a page list (QBGLAY).

This is the number of coupling facility reads to retrieve a changed page from the group buffer pool as a result of feedback from the request to register a page list (DB2 4.1 and later).

DB2 prefetch can do this only if the member is running on MVS 5.2 or later and if the group buffer pool is allocated in a coupling facility with CFLEVEL=2 or above.

REGISTER PAGE LIST-RD CLEAN

Number of coupling facility requests to retrieve a clean page for a page list (QBGLAZ).

This is the number of coupling facility reads to retrieve a clean page from the group buffer pool as a result of feedback from the request to register a page list (DB2 4.1 and later).

DB2 prefetch can do this only if the member is running on MVS 5.2 or later and if the group buffer pool is allocated in a coupling facility with CFLEVEL=2 or above.

EXPLICIT CROSS INVALIDATES

Number of explicit cross-invalidations (QBGLEX).

(DB2 6.1 and later)

DUPLEX-WRITE REQUEST

Number of coupling facility requests to write changed pages to the secondary group buffer pool for duplexing (QBGL2W).

(DB2 6.1 and later)

DUPLEX-WRITE FAIL

Number of coupling facility requests to write changed pages to the secondary group buffer pool for duplexing that failed due to a lack of storage in the coupling facility (QBGL2F).

(DB2 6.1 and later)

DUPLEX-DELETE NAME LIST

Number of group buffer pool requests to the secondary group buffer pool to delete a list of pages after they have been castout from the primary group buffer pool (QBGL2D).

(DB2 6.1 and later)

DUPLEX-DELETE NAME

Number of group buffer pool requests to delete a page from the secondary group buffer pool (QBGL2N).

These requests are issued by the group buffer pool structure owner to delete orphaned data entries in the secondary as part of the garbage collection logic.

(DB2 6.1 and later)

DUPLEX-READ CASTOUT STATS

Number of coupling facility requests to read the castout statistics for the secondary group buffer pool (QBGL2R).

These requests are issued by the group buffer pool structure owner to check for orphaned data entries in the secondary.

(DB2 6.1 and later)

## ROUTINES

These fields apply only to DB2 4.1 and later.

SPROC-CALL STATEMENTS
Number of CALL statements executed (QXCALL).

This counter is incremented every time an application program issues an SQL CALL statement to execute a DB2 stored procedure.

SPROC-ABENDS
Number of stored procedure abends (QXCALLAB).

This counter is incremented when a stored procedure abends.

Large numbers here indicate that one or more stored procedures have fatal errors.

SPROC-TIMEOUTS
Number of CALL statements timed out (QXCALLTO).

This counter is incremented when the Stored Procedures Address Space does not have a TCB available within the time limit specified in the DSNZPARM TIMEOUT parameter to schedule the execution of an SQL CALL statement.

Appropriate fixes include

- Reduce the amount of CPU service units in other stored procedures to get better throughput.
- Stop stored procedures no longer needed.
- Increase the number of TCBs available in the Stored Procedures Address Space (by increasing the NUMTCB parameter in the Stored Procedures Address Space JCL).

SPROC-REJECTS
Number of CALL statements rejected because procedure was stopped (QXCALLRJ).

This counter is incremented when an SQL application CALLs a stored procedure which has been stopped by the operator or system administrator.

**Tuning Tip:** If this happens frequently, be sure the procedure is started in normal mode.

TRIGGER-STMT ACTIVATED
Number of times a statement trigger is activated (QXSTTRG).

(DB2 6.1 and later)

TRIGGER-ROW ACTIVATED
Number of times a row trigger is activated (QXROWTRG).

(DB2 6.1 and later)

TRIGGER-SQL ERRORS
Number of times an SQL error occurred during execution of a triggered action (QXTRGERR).

(DB2 6.1 and later)

UDF -EXECUTIONS
Number of user-defined functions executed (QXCAUD).

(DB2 6.1 and later)

UDF -ABENDS
Number of times a user-defined function abended (QXCAUDAB).

(DB2 6.1 and later)

UDF -TIMEOUTS
Number of times a user-defined function timed out waiting to be scheduled (QXCAUDTO).

(DB2 6.1 and later)

UDF -REJECTS
Number of times a user-defined function was rejected (QXCAUDRJ).

(DB2 6.1 and later)

# Statistics Short Report—DDF

The fields in this section are organized in the following categories:

**Note:** This page is produced only if distributed work is done.

## DRDA REMOTE LOCATION

REMOTE LOCATION NAME
Location name of remote site (QLSTLOCN).

Location name of the remote site involved in distributed work with this site, either as a server or requester.

Statistics for all remote locations accessed by DRDA protocol are grouped under the location name “DRDA REMOTE LOCS”. For DB2 private protocol, statistics are gathered for each remote location.

BEGIN INTERVAL DATE
Beginning date and time of the reporting interval in which DDF work was processed.

END INTERVAL DATE
Ending date and time of the reporting interval in which DDF work was processed.

## DISTRIBUTED ACTIVITY

TRANSACTIONS RECEIVED

Number of transactions received (QLSTTRNR).

This counter is incremented at the server location and reflects the number of threads requested by the requester.

**Tuning Tip:** This counter is useful only at the server location since DB2 maintains local site autonomy based on the number of active and inactive threads in DSNZPARM.

TRANSACTIONS SENT

Number of transactions sent (QLSTTRNS).

This counter is incremented at the requester location and reflects the total number of threads requested from the server location. Each one allocates a DBAT thread.

**Tuning Tip:** This data is useful only at the requester location since DB2 maintains local site autonomy based on the number of active and inactive threads in DSNZPARM.

CONVERSATIONS RECEIVED

Number of conversations initiated to the server (QLSTCNVR).

This counter is incremented at the server location for each conversation the server has successfully retrieved.

**Tuning Tip:** Performance tuning is the art of getting as many data rows possible in as few conversations as possible.

CONVERSATIONS SENT

Number of conversations initiated from the requester location (QLSTCNVS).

This counter is incremented at the requester location and reflects the number of specific conversations sent to all server locations.

Conversations are specific instances of transmitting data back and forth. Sessions are logical connections between requesters and servers. It is analogous to a telephone call in which the call is a session and each party talking is a conversation.

CONVERSATIONS QUEUED

Number of conversations queued (QLSTCNVQ).

This counter is incremented at the requester location every time a conversation request to VTAM is initiated but cannot yet be started.

**Tuning Tip:** The way to avoid queuing conversational requests from a DB2 is to increase the conversation limits column in the SYSIBM.SYSLUMODE table to a larger number. This should be done in conjunction with the VTAM system programmer to determine if the network can handle increased numbers of conversations.

SQL BOUND FOR REMOTE ACCESS

Number of SQL statements bound for remote access (QLSTRBND).

This value is incremented at the requester location each time a system-directed SQL statement is encountered (three part name, no CONNECT statement, DB2 private protocol), where dynamic requests occur at the server.

One of the expensive features of the DB2 private protocol is that the SQL statement is dynamically run on the server site and thus incurs extra overhead at the server.

COMMITS RECEIVED

Number of phase 1 and phase 2 commit requests received (QLSTCOMR+QLSTCRRC).

COMMITS SENT

Number of phase 1 and phase 2 commit requests sent (QLSTCOMS+QLSTCRSE).

SQL STATEMENTS RECEIVED

Number of SQL statements received from the requester (QLSTSQLR).

This counter is incremented at the server location when an SQL statement is received.

**Tuning Tip:** This information, along with other message information, can help determine the relative efficiency of the distributed applications and can, in conjunction with other data items, help in the network tuning of distributed applications.

SQL STATEMENTS SENT

Number of SQL statements sent to the remote server (QLSTSQLS).

This counter is incremented at the requester location statistics when an SQL statement is sent.

**Tuning Tip:** This information, along with other message information, can help determine the relative efficiency of the distributed applications and can, in conjunction with other data items, help in the network tuning of distributed applications.

ROWS RECEIVED

Number of rows received from the server (QLSTROWR).

This counter is incremented at the requester and reflects the number of rows retrieved from the server.

**Tuning Tip:** From a performance perspective, the ideal is to retrieve as many rows in as few conversational messages as possible. This is accomplished using a feature of DB2 called block fetch. Block fetch is used when a cursor is determined unambiguously to be a read-only cursor. The use of the FOR FETCH ONLY clause along with the bind option of CURRENTDATA(NO) can help DB2 achieve this goal. Block fetch is always most effective with system-directed SQL (three-part names without a CONNECT statement, sometimes called DB2 private protocols). In using private protocols, DB2 fits as many rows as possible into a single 32K message. The typical overhead in distributed performance is network bandwidth. If the application can send fewer messages, the distributed application is more efficient.

ROWS SENT

Number of rows sent to the requester (QLSTROWS).

This counter is incremented at the server for each row sent to a remote requester.

This number, combined with the number of messages (conversations), can be used in determining whether efficiency techniques such as block fetch are being used.

MESSAGES RECEIVED

Number of messages received from the remote location (QLSTMSGR).

This counter is incremented at the receiving location reflecting the number of messages this location received.

The count of messages received can often be more than the number of messages sent because of the way distributed SQL statements are processed.

MESSAGES SENT

Number of messages sent to the remote location (QLSTMSGS).

This counter is incremented at the originating location. It reflects a count of each VTAM message.

A message, in VTAM terms is a group of characters and control bit sequences which are transmitted as a single unit of data, including header information and data.

ROLLBACKS RECEIVED

Number of phase 1 and phase 2 rollback requests received (QLSTABRR+QLSTBKRC).

ROLLBACKS SENT

Number of phase 1 and phase 2 rollback requests sent (QLSTABRS+QLSTBKSE).

BYTES RECEIVED

Number of bytes received from the server (QLSTBYTR).

This counter is incremented at the requester reflecting the number of bytes retrieved by the server.

The number of bytes received may be greater than the number sent since the overhead of network transmission tends to add message information in the form of additional VTAM header blocks.

BYTES SENT

Number of bytes sent to the requester (QLSTBYTS).

This count represents the number of bytes sent by the server. This counter is incremented at the server location.

**Tuning Tip:** The number of bytes per message can help the network tuner decide the appropriate RUSIZE and pacing to fit the network.

BLOCKS RECEIVED - BLK FETCH

Number of blocks received using block fetch (QLSTBRBF).

This counter, incremented at the requester location, indicates how many blocks were received to process remote requesters.

**Tuning Tip:** Each block costs VTAM transmission time. The more rows that can be transmitted per block, the better performance for distributed transactions will be.

BLOCKS SENT - BLK FETCH

Number of blocks transmitted using block fetch (QLSTBTBF).

This counter, incremented at the server location, indicates how many blocks were sent to remote requesters.

**Tuning Tip:** Each block costs VTAM transmission time. The more rows that can be transmitted per block, the better performance for distributed transactions will be.

MESSAGES IN BUFFER

Number of rows transmitted or received using block fetch (QLSTBROW).

This counter, incremented at both requester and server, indicates the number of rows in each block of a block fetch operation.

**Tuning Tip:** Generally speaking, the higher the number of rows per block, the better performance for distributed transactions will be.

SWITCH LIMITED BLK PROTOCOL

Number of switches from continuous to limited block fetch (QLSTCBLB).

This counter is incremented at the requester location and applies only to system-directed applications (three-part names, no CONNECT statement, sometimes called private protocols). It reflects the number of times asynchronous block fetch had to be reduced to a more limited form of block fetch because of a lack of network resources.

**Tuning Tip:** This situation occurs when too many conversations are occurring at this site for VTAM to handle the asynchronous fetch of data. It needs to be resolved by the MVS and VTAM systems programmers to allow more conversations to take place.

# Chapter 10.  Statistics Long Report

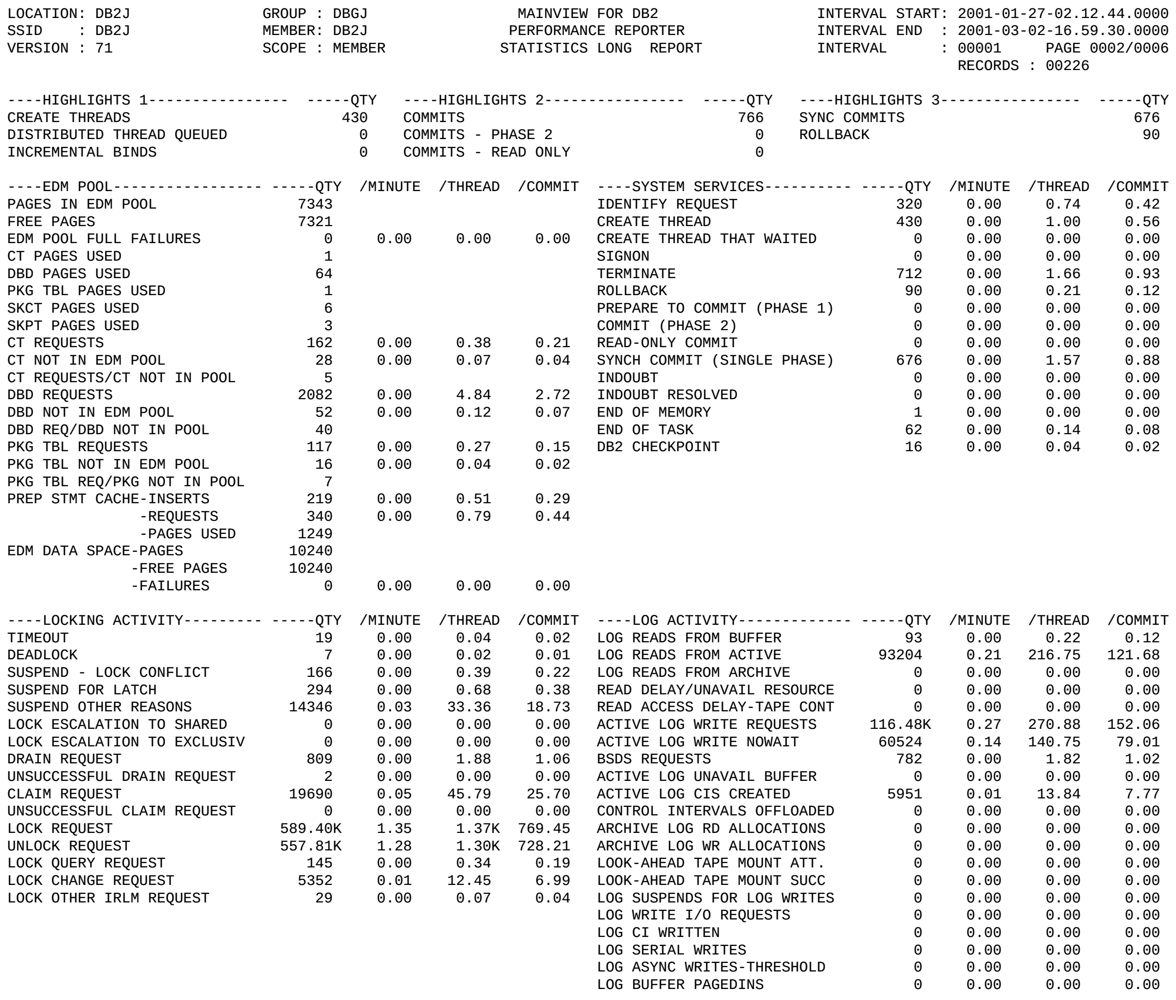
This chapter describes the statistics long report, as shown in Figure 23–Figure 29. It has six pages per summary key, followed by a page for DDF information, if present.

```
LOCATION: DB2J               GROUP : DBGJ                 MAINVIEW FOR DB2                  INTERVAL START: 2001-01-27-02.12.44.0000
SSID    : DB2J               MEMBER: DB2J                PERFORMANCE REPORTER               INTERVAL END  : 2001-03-02-16.59.30.0000
VERSION : 71                 SCOPE : MEMBER             STATISTICS LONG  REPORT             INTERVAL      : 00001     PAGE 0001/0006
                                                                                                             RECORDS : 00226

----HIGHLIGHTS 1----------------  -----QTY   ----HIGHLIGHTS 2---------------  -----QTY   ----HIGHLIGHTS 3---------------- -----QTY
CREATE THREADS                         430   COMMITS                                766   SYNC COMMITS                           676
DISTRIBUTED THREAD QUEUED                0   COMMITS - PHASE 2                        0   ROLLBACK                                90
INCREMENTAL BINDS                        0   COMMITS - READ ONLY                      0

----SQL DML------------------ -----QTY  /MINUTE  /THREAD  /COMMIT  ----SQL DCL----------------- -----QTY  /MINUTE  /THREAD  /COMMIT
SELECT                               0     0.00     0.00     0.00  LOCK TABLE                          0     0.00     0.00     0.00
INSERT                              18     0.00     0.04     0.02  GRANT                              21     0.00     0.05     0.03
UPDATE                              50     0.00     0.12     0.07  REVOKE                              0     0.00     0.00     0.00
DELETE                              38     0.00     0.09     0.05  SET CURRENT SQLID                   0     0.00     0.00     0.00
PREPARE                            320     0.00     0.74     0.42  SET HOST VARIABLE                   0     0.00     0.00     0.00
DESCRIBE                            17     0.00     0.04     0.02  SET CURRENT DEGREE                  0     0.00     0.00     0.00
DESCRIBE TABLE                       0     0.00     0.00     0.00  SET CURRENT RULES                   0     0.00     0.00     0.00
OPEN                               196     0.00     0.46     0.26  SET CURRENT PATH                    0     0.00     0.00     0.00
FETCH                            75352     0.17   175.24    98.37  CONNECT TYPE 1                      0     0.00     0.00     0.00
CLOSE                              193     0.00     0.45     0.25  CONNECT TYPE 2                      6     0.00     0.01     0.01
TOTAL                            76184     0.17   177.17    99.46  SET CONNECTION                      0     0.00     0.00     0.00
                                                                   RELEASE                             0     0.00     0.00     0.00
----SQL DDL------------------ -----QTY  /MINUTE  /THREAD  /COMMIT  CALL                                0     0.00     0.00     0.00
ALTER DATABASE                       0     0.00     0.00     0.00  ASSOCIATE LOCATOR                   0     0.00     0.00     0.00
ALTER STORAGE GROUP                  0     0.00     0.00     0.00  HOLD LOCATOR                        0     0.00     0.00     0.00
ALTER INDEX                          0     0.00     0.00     0.00  FREE LOCATOR                        0     0.00     0.00     0.00
ALTER TABLE                         20     0.00     0.05     0.03  ALLOCATE CURSOR                     0     0.00     0.00     0.00
ALTER TABLESPACE                     4     0.00     0.01     0.01  TOTALS                             54     0.00     0.13     0.07
ALTER PROCEDURE                      0     0.00     0.00     0.00  COMMENT ON                          0     0.00     0.00     0.00
ALTER FUNCTION                       0     0.00     0.00     0.00  RENAME TABLE                        0     0.00     0.00     0.00
CREATE ALIAS                         0     0.00     0.00     0.00  LABEL ON                            0     0.00     0.00     0.00
CREATE DATABASE                      1     0.00     0.00     0.00
CREATE INDEX                        33     0.00     0.08     0.04  ----OPTIMIZATION ------------ -----QTY  /MINUTE  /THREAD  /COMMIT
CREATE STORAGE GROUP                 1     0.00     0.00     0.00  PREPARE - SATISFIED               107     0.00     0.25     0.14
CREATE SYNONYM                       2     0.00     0.00     0.00          - FAILED SEARCH           233     0.00     0.54     0.30
CREATE TABLE                        33     0.00     0.08     0.04          - IMPLICIT                  0     0.00     0.00     0.00
CREATE TABLESPACE                   29     0.00     0.07     0.04          - FROM CACHE                0     0.00     0.00     0.00
CREATE VIEW                          8     0.00     0.02     0.01          - DISCARDED MAX             0     0.00     0.00     0.00
CREATE TEMP TABLE                    0     0.00     0.00     0.00          - DROP/ALTER/REVOKE        12     0.00     0.03     0.02
CREATE AUX TABLE                     0     0.00     0.00     0.00
CREATE PROCEDURE                     0     0.00     0.00     0.00
CREATE FUNCTION                      0     0.00     0.00     0.00  ----RID LIST----------------- -----QTY  /MINUTE  /THREAD  /COMMIT
CREATE TRIGGER                       0     0.00     0.00     0.00  RID CURRENT BLOCKS ALLOCATE         0
CREATE DISTINCT TYPE                 0     0.00     0.00     0.00  RID HIGH BLOCKS ALLOCATED           0
DROP ALIAS                           0     0.00     0.00     0.00  RID TERM OVER RDS LIMIT             0     0.00     0.00     0.00
DROP DATABASE                        0     0.00     0.00     0.00  RID TERM TOO MANY CONCURR           0     0.00     0.00     0.00
DROP INDEX                           0     0.00     0.00     0.00  RID TERM - NO STORAGE               0     0.00     0.00     0.00
DROP STORAGE GROUP                   0     0.00     0.00     0.00  RID TERM - OVER DM LIMIT            0     0.00     0.00     0.00
DROP SYNONYM                         0     0.00     0.00     0.00
DROP TABLE                           0     0.00     0.00     0.00  ----OPEN/CLOSE ACTIVITY------ -----QTY  /MINUTE  /THREAD  /COMMIT
DROP TABLESPACE                     18     0.00     0.04     0.02  SLOW CLOSE DATASETS               542
DROP VIEW                            0     0.00     0.00     0.00  DATASETS CURRENTLY OPEN           543
DROP PACKAGE                         0     0.00     0.00     0.00  IN-USE DATA SETS                    1
DROP PROCEDURE                       0     0.00     0.00     0.00  MAX SLOW CLOSE DATASETS           542
DROP FUNCTION                        0     0.00     0.00     0.00  MAX D.S. OPEN CONCURRENTLY        543
DROP TRIGGER                         0     0.00     0.00     0.00  D.S. CLOSED (THRESHOLD)            56     0.00     0.13     0.07
DROP DISTINCT TYPE                   0     0.00     0.00     0.00  D.S. CONVERTED - RW TO RO         371     0.00     0.86     0.48
TOTAL                              149     0.00     0.35     0.19  REOPEN FROM SLOW CLOSE          14489     0.03    33.70    18.92
```

Figure 23.  Statistics Long Report—Page 1

For a complete description of each section, see

- “HIGHLIGHTS” on page 254
- “SQL DML” on page 256
- “SQL DDL” on page 257
- “SQL DCL” on page 258
- “OPTIMIZATION” on page 261
- “RID LIST” on page 262
- “OPEN/CLOSE ACTIVITY” on page 263

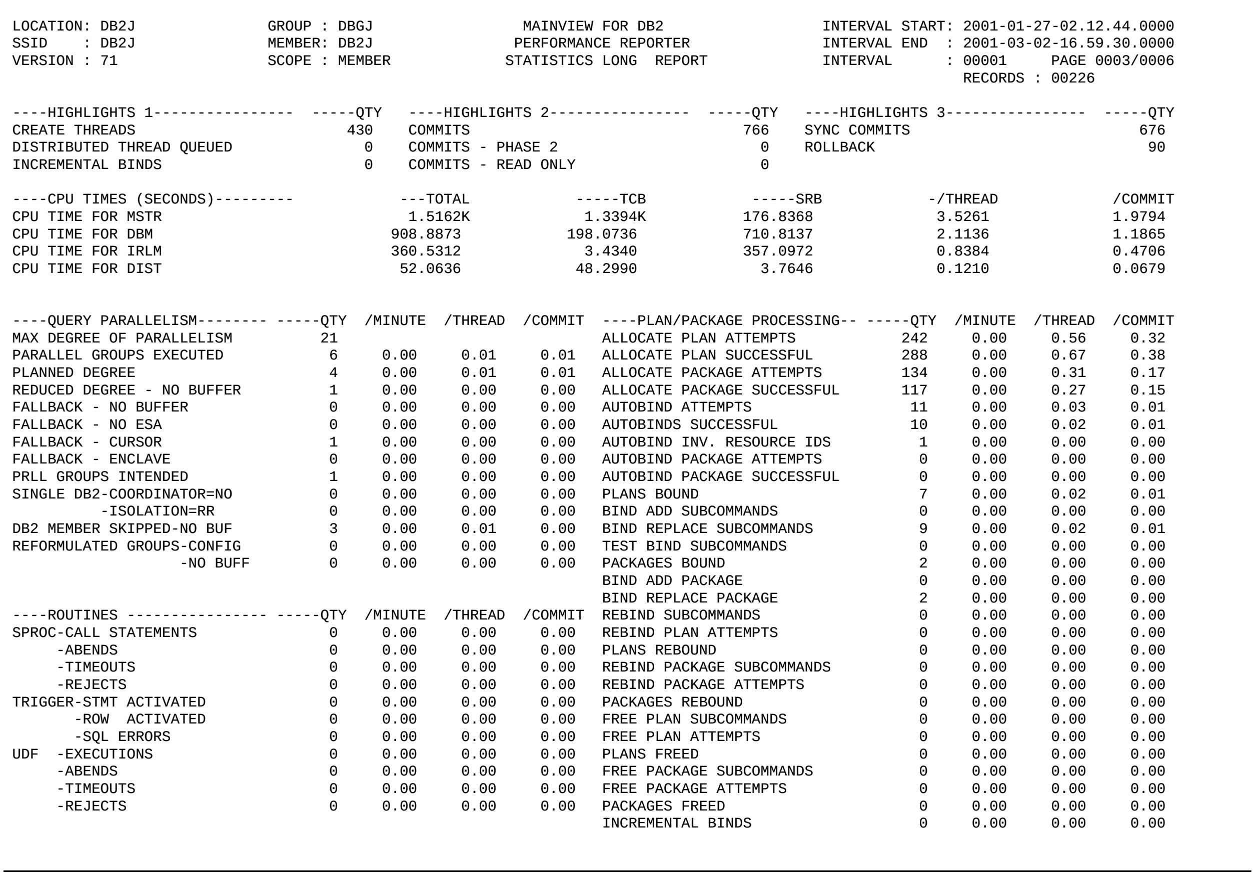


```
LOCATION: DB2J              GROUP : DBGJ               MAINVIEW FOR DB2              INTERVAL START: 2001-01-27-02.12.44.0000
SSID    : DB2J              MEMBER: DB2J              PERFORMANCE REPORTER           INTERVAL END  : 2001-03-02-16.59.30.0000
VERSION : 71                SCOPE : MEMBER           STATISTICS LONG  REPORT         INTERVAL      : 00001     PAGE 0002/0006
                                                                                                    RECORDS : 00226

----HIGHLIGHTS 1----------------  -----QTY   ----HIGHLIGHTS 2----------------  -----QTY   ----HIGHLIGHTS 3----------------  -----QTY
CREATE THREADS                          430    COMMITS                                766    SYNC COMMITS                          676
DISTRIBUTED THREAD QUEUED                 0    COMMITS - PHASE 2                        0    ROLLBACK                               90
INCREMENTAL BINDS                         0    COMMITS - READ ONLY                      0

----EDM POOL----------------- -----QTY  /MINUTE  /THREAD  /COMMIT  ----SYSTEM SERVICES---------- -----QTY  /MINUTE  /THREAD  /COMMIT
PAGES IN EDM POOL                 7343                              IDENTIFY REQUEST                   320     0.00     0.74     0.42
FREE PAGES                        7321                              CREATE THREAD                      430     0.00     1.00     0.56
EDM POOL FULL FAILURES               0     0.00     0.00     0.00   CREATE THREAD THAT WAITED            0     0.00     0.00     0.00
CT PAGES USED                        1                              SIGNON                               0     0.00     0.00     0.00
DBD PAGES USED                      64                              TERMINATE                          712     0.00     1.66     0.93
PKG TBL PAGES USED                   1                              ROLLBACK                            90     0.00     0.21     0.12
SKCT PAGES USED                      6                              PREPARE TO COMMIT (PHASE 1)          0     0.00     0.00     0.00
SKPT PAGES USED                      3                              COMMIT (PHASE 2)                     0     0.00     0.00     0.00
CT REQUESTS                        162     0.00     0.38     0.21   READ-ONLY COMMIT                     0     0.00     0.00     0.00
CT NOT IN EDM POOL                  28     0.00     0.07     0.04   SYNCH COMMIT (SINGLE PHASE)        676     0.00     1.57     0.88
CT REQUESTS/CT NOT IN POOL           5                              INDOUBT                              0     0.00     0.00     0.00
DBD REQUESTS                      2082     0.00     4.84     2.72   INDOUBT RESOLVED                     0     0.00     0.00     0.00
DBD NOT IN EDM POOL                 52     0.00     0.12     0.07   END OF MEMORY                        1     0.00     0.00     0.00
DBD REQ/DBD NOT IN POOL             40                              END OF TASK                         62     0.00     0.14     0.08
PKG TBL REQUESTS                   117     0.00     0.27     0.15   DB2 CHECKPOINT                      16     0.00     0.04     0.02
PKG TBL NOT IN EDM POOL             16     0.00     0.04     0.02
PKG TBL REQ/PKG NOT IN POOL          7
PREP STMT CACHE-INSERTS            219     0.00     0.51     0.29
               -REQUESTS           340     0.00     0.79     0.44
               -PAGES USED        1249
EDM DATA SPACE-PAGES             10240
              -FREE PAGES        10240
              -FAILURES              0     0.00     0.00     0.00

----LOCKING ACTIVITY--------- -----QTY  /MINUTE  /THREAD  /COMMIT  ----LOG ACTIVITY------------ -----QTY  /MINUTE  /THREAD  /COMMIT
TIMEOUT                             19     0.00     0.04     0.02   LOG READS FROM BUFFER               93     0.00     0.22     0.12
DEADLOCK                             7     0.00     0.02     0.01   LOG READS FROM ACTIVE            93204     0.21   216.75   121.68
SUSPEND - LOCK CONFLICT            166     0.00     0.39     0.22   LOG READS FROM ARCHIVE               0     0.00     0.00     0.00
SUSPEND FOR LATCH                  294     0.00     0.68     0.38   READ DELAY/UNAVAIL RESOURCE          0     0.00     0.00     0.00
SUSPEND OTHER REASONS            14346     0.03    33.36    18.73   READ ACCESS DELAY-TAPE CONT          0     0.00     0.00     0.00
LOCK ESCALATION TO SHARED            0     0.00     0.00     0.00   ACTIVE LOG WRITE REQUESTS      116.48K     0.27   270.88   152.06
LOCK ESCALATION TO EXCLUSIV          0     0.00     0.00     0.00   ACTIVE LOG WRITE NOWAIT          60524     0.14   140.75    79.01
DRAIN REQUEST                      809     0.00     1.88     1.06   BSDS REQUESTS                      782     0.00     1.82     1.02
UNSUCCESSFUL DRAIN REQUEST           2     0.00     0.00     0.00   ACTIVE LOG UNAVAIL BUFFER            0     0.00     0.00     0.00
CLAIM REQUEST                    19690     0.05    45.79    25.70   ACTIVE LOG CIS CREATED            5951     0.01    13.84     7.77
UNSUCCESSFUL CLAIM REQUEST           0     0.00     0.00     0.00   CONTROL INTERVALS OFFLOADED          0     0.00     0.00     0.00
LOCK REQUEST                   589.40K     1.35    1.37K   769.45   ARCHIVE LOG RD ALLOCATIONS           0     0.00     0.00     0.00
UNLOCK REQUEST                 557.81K     1.28    1.30K   728.21   ARCHIVE LOG WR ALLOCATIONS           0     0.00     0.00     0.00
LOCK QUERY REQUEST                 145     0.00     0.34     0.19   LOOK-AHEAD TAPE MOUNT ATT.           0     0.00     0.00     0.00
LOCK CHANGE REQUEST               5352     0.01    12.45     6.99   LOOK-AHEAD TAPE MOUNT SUCC           0     0.00     0.00     0.00
LOCK OTHER IRLM REQUEST             29     0.00     0.07     0.04   LOG SUSPENDS FOR LOG WRITES          0     0.00     0.00     0.00
                                                                    LOG WRITE I/O REQUESTS               0     0.00     0.00     0.00
                                                                    LOG CI WRITTEN                       0     0.00     0.00     0.00
                                                                    LOG SERIAL WRITES                    0     0.00     0.00     0.00
                                                                    LOG ASYNC WRITES-THRESHOLD           0     0.00     0.00     0.00
                                                                    LOG BUFFER PAGEDINS                  0     0.00     0.00     0.00
```

Figure 24. Statistics Long Report—Page 2

For a complete description of each section, see

- “EDM POOL” on page 265
- “SYSTEM SERVICES” on page 269
- “LOCKING ACTIVITY” on page 272
- “LOG ACTIVITY” on page 275



```
LOCATION: DB2J               GROUP : DBGJ                 MAINVIEW FOR DB2                  INTERVAL START: 2001-01-27-02.12.44.0000
SSID    : DB2J               MEMBER: DB2J                PERFORMANCE REPORTER               INTERVAL END  : 2001-03-02-16.59.30.0000
VERSION : 71                 SCOPE : MEMBER             STATISTICS LONG  REPORT             INTERVAL      : 00001     PAGE 0003/0006
                                                                                                            RECORDS : 00226

----HIGHLIGHTS 1----------------  -----QTY   ----HIGHLIGHTS 2----------------  -----QTY   ----HIGHLIGHTS 3----------------  -----QTY
CREATE THREADS                        430    COMMITS                               766    SYNC COMMITS                          676
DISTRIBUTED THREAD QUEUED               0    COMMITS - PHASE 2                       0    ROLLBACK                               90
INCREMENTAL BINDS                       0    COMMITS - READ ONLY                     0

----CPU TIMES (SECONDS)---------            ---TOTAL            -----TCB            -----SRB            -/THREAD             /COMMIT
CPU TIME FOR MSTR                            1.5162K             1.3394K          176.8368              3.5261              1.9794
CPU TIME FOR DBM                           908.8873            198.0736           710.8137              2.1136              1.1865
CPU TIME FOR IRLM                          360.5312              3.4340           357.0972              0.8384              0.4706
CPU TIME FOR DIST                           52.0636             48.2990             3.7646              0.1210              0.0679


----QUERY PARALLELISM-------- -----QTY  /MINUTE  /THREAD  /COMMIT  ----PLAN/PACKAGE PROCESSING-- -----QTY  /MINUTE  /THREAD  /COMMIT
MAX DEGREE OF PARALLELISM          21                              ALLOCATE PLAN ATTEMPTS            242     0.00     0.56     0.32
PARALLEL GROUPS EXECUTED            6     0.00     0.01     0.01   ALLOCATE PLAN SUCCESSFUL          288     0.00     0.67     0.38
PLANNED DEGREE                      4     0.00     0.01     0.01   ALLOCATE PACKAGE ATTEMPTS         134     0.00     0.31     0.17
REDUCED DEGREE - NO BUFFER          1     0.00     0.00     0.00   ALLOCATE PACKAGE SUCCESSFUL       117     0.00     0.27     0.15
FALLBACK - NO BUFFER                0     0.00     0.00     0.00   AUTOBIND ATTEMPTS                  11     0.00     0.03     0.01
FALLBACK - NO ESA                   0     0.00     0.00     0.00   AUTOBINDS SUCCESSFUL               10     0.00     0.02     0.01
FALLBACK - CURSOR                   1     0.00     0.00     0.00   AUTOBIND INV. RESOURCE IDS          1     0.00     0.00     0.00
FALLBACK - ENCLAVE                  0     0.00     0.00     0.00   AUTOBIND PACKAGE ATTEMPTS           0     0.00     0.00     0.00
PRLL GROUPS INTENDED                1     0.00     0.00     0.00   AUTOBIND PACKAGE SUCCESSFUL         0     0.00     0.00     0.00
SINGLE DB2-COORDINATOR=NO           0     0.00     0.00     0.00   PLANS BOUND                         7     0.00     0.02     0.01
          -ISOLATION=RR             0     0.00     0.00     0.00   BIND ADD SUBCOMMANDS                0     0.00     0.00     0.00
DB2 MEMBER SKIPPED-NO BUF           3     0.00     0.01     0.00   BIND REPLACE SUBCOMMANDS            9     0.00     0.02     0.01
REFORMULATED GROUPS-CONFIG          0     0.00     0.00     0.00   TEST BIND SUBCOMMANDS               0     0.00     0.00     0.00
                   -NO BUFF         0     0.00     0.00     0.00   PACKAGES BOUND                      2     0.00     0.00     0.00
                                                                   BIND ADD PACKAGE                    0     0.00     0.00     0.00
                                                                   BIND REPLACE PACKAGE                2     0.00     0.00     0.00
----ROUTINES ---------------- -----QTY  /MINUTE  /THREAD  /COMMIT  REBIND SUBCOMMANDS                  0     0.00     0.00     0.00
SPROC-CALL STATEMENTS               0     0.00     0.00     0.00   REBIND PLAN ATTEMPTS                0     0.00     0.00     0.00
     -ABENDS                        0     0.00     0.00     0.00   PLANS REBOUND                       0     0.00     0.00     0.00
     -TIMEOUTS                      0     0.00     0.00     0.00   REBIND PACKAGE SUBCOMMANDS          0     0.00     0.00     0.00
     -REJECTS                       0     0.00     0.00     0.00   REBIND PACKAGE ATTEMPTS             0     0.00     0.00     0.00
TRIGGER-STMT ACTIVATED              0     0.00     0.00     0.00   PACKAGES REBOUND                    0     0.00     0.00     0.00
       -ROW  ACTIVATED              0     0.00     0.00     0.00   FREE PLAN SUBCOMMANDS               0     0.00     0.00     0.00
       -SQL ERRORS                  0     0.00     0.00     0.00   FREE PLAN ATTEMPTS                  0     0.00     0.00     0.00
UDF  -EXECUTIONS                    0     0.00     0.00     0.00   PLANS FREED                         0     0.00     0.00     0.00
     -ABENDS                        0     0.00     0.00     0.00   FREE PACKAGE SUBCOMMANDS            0     0.00     0.00     0.00
     -TIMEOUTS                      0     0.00     0.00     0.00   FREE PACKAGE ATTEMPTS               0     0.00     0.00     0.00
     -REJECTS                       0     0.00     0.00     0.00   PACKAGES FREED                      0     0.00     0.00     0.00
                                                                   INCREMENTAL BINDS                   0     0.00     0.00     0.00
```

Figure 25. Statistics Long Report—Page 3

For a complete description of each section, see

- “CPU TIMES (SECONDS)” on page 279
- “QUERY PARALLELISM” on page 280
- “PLAN/PACKAGE PROCESSING” on page 285
- “SQL DCL” on page 258

```
LOCATION: DB2J               GROUP : DBGJ                 MAINVIEW FOR DB2                  INTERVAL START: 2001-01-27-02.12.44.0000
SSID    : DB2J               MEMBER: DB2J                PERFORMANCE REPORTER               INTERVAL END  : 2001-03-02-16.59.30.0000
VERSION : 71                 SCOPE : MEMBER             STATISTICS LONG  REPORT             INTERVAL      : 00001     PAGE 0004/0006
                                                                                                            RECORDS : 00226

----HIGHLIGHTS 1----------------  -----QTY   ----HIGHLIGHTS 2----------------  -----QTY   ----HIGHLIGHTS 3----------------  -----QTY
CREATE THREADS                        430    COMMITS                               766    SYNC COMMITS                          676
DISTRIBUTED THREAD QUEUED               0    COMMITS - PHASE 2                       0    ROLLBACK                               90
INCREMENTAL BINDS                       0    COMMITS - READ ONLY                     0

----GLOBAL BUFFERPOOL ------- -----QTY  /MINUTE  /THREAD  /COMMIT  --GLOBAL LOCKING ACTIVITY --- -----QTY  /MINUTE  /THREAD  /COMMIT
SYNC RD INV BUFF -WITH DATA        58     0.00     0.13     0.08   PLOCK REQUESTS-LOCK              5001     0.01    11.63     6.53
           - NO DATA               12     0.00     0.03     0.02                 -CHANGE             958     0.00     2.23     1.25
SYNC RD NOT FOUND-WITH DATA        18     0.00     0.04     0.02                 -UNLOCK            1224     0.00     2.85     1.60
           - NO DATA              128     0.00     0.30     0.17   XES SYNC REQ  -LOCK            444.65K    1.02     1.03K  580.48
ASYNC READ - DATA RETURNED          0     0.00     0.00     0.00                 -CHANGE             690     0.00     1.60     0.90
           - NO DATA              196     0.00     0.46     0.26                 -UNLOCK          448.52K    1.03     1.04K  585.54
SYNC PAGES WRITTEN -CHANGED      1158     0.00     2.69     1.51   SUSPENDS      -IRLM              5494     0.01    12.78     7.17
                   -CLEAN           0     0.00     0.00     0.00                 -XES                139     0.00     0.32     0.18
ASYNC PAGES WRITTEN-CHANGED       205     0.00     0.48     0.27                 -FALSE             2356     0.01     5.48     3.08
                   -CLEAN           0     0.00     0.00     0.00   GLOBAL LOCK CONTENTION(%)           0
CASTOUT -PAGES WRITTEN            783     0.00     1.82     1.02   FALSE CONTENTION(%)                29
        -NO ENGINE                  0     0.00     0.00     0.00   INCOMPATIBLE RETAINED LOCK          0     0.00     0.00     0.00
        -CLASS THRESHOLD           10     0.00     0.02     0.01   NOTIFY MESSAGES -SENT           20522     0.05    47.73    26.79
        -GBP THRESHOLD              4     0.00     0.01     0.01   NOTIFY MESSAGES -RECEIVED        1986     0.00     4.62     2.59
NO WRITE ENGINE                     0     0.00     0.00     0.00   XES ASYNC REQ -RESOURCES           14     0.00     0.03     0.02
READ FAIL  -STORAGE                 0     0.00     0.00     0.00   NEGOTIATE -PGSET/PARTITION        466     0.00     1.08     0.61
WRITE FAIL -STORAGE                 0     0.00     0.00     0.00             -PAGE PLOCK              38     0.00     0.09     0.05
RD STG STATS(5.1)/OTHER(4.1         0     0.00     0.00     0.00             -OTHER PLOCK            132     0.00     0.31     0.17
GBP CHECKPOINT                    420     0.00     0.98     0.55             -PLOCK CHANGE REQ       512     0.00     1.19     0.67
GBP REBUILD                         0     0.00     0.00     0.00   PLOCK/NOTIFY -MAX ENGINES         500
UNLOCK CASTOUT                    325     0.00     0.76     0.42                -NO ENGINE             0
READ CASTOUT CLASS                576     0.00     1.34     0.75
READ CASTOUT STATISTICS           425     0.00     0.99     0.55   ----AUTHORIZATION MANAGEMENT- -----QTY  /MINUTE  /THREAD  /COMMIT
DELETE DIR/DATA ENTRIES          1196     0.00     2.78     1.56   PLAN-AUTHORIZATION CHECKS        1659     0.00     3.86     2.17
READ DIRECTORY INFORMATION          0     0.00     0.00     0.00       -SUCC CHECKS                 1638     0.00     3.81     2.14
REGISTER PAGE                     884     0.00     2.06     1.15       -SUCC CHECKS -PUBLIC           86     0.00     0.20     0.11
UNREGISTER PAGE                   189     0.00     0.44     0.25       -SUCC CHECKS -CACHE            93     0.00     0.22     0.12
REGISTER PAGE LIST                118     0.00     0.27     0.15   PKG -SUCC CHECKS -CACHE            91     0.00     0.21     0.12
REGISTER PAGE LIST-RD CHNGE         0     0.00     0.00     0.00       -UNSUCC CHECKS -CACHE          29     0.00     0.07     0.04
REGISTER PAGE LIST-RD CLEAN         0     0.00     0.00     0.00       -SUCC CHECKS -PUBLIC            0     0.00     0.00     0.00
EXPLICIT CROSS INVALIDATES          0     0.00     0.00     0.00   OVERWROTE AUTHID -CACHE             0     0.00     0.00     0.00
DUPLEX-WRITE REQUEST                0     0.00     0.00     0.00   OVERWROTE PKG/COLLID -CACHE         0     0.00     0.00     0.00
      -WRITE FAIL                   0     0.00     0.00     0.00   ROUTINE CACHE-SUCCESS               0     0.00     0.00     0.00
      -DELETE NAME LIST             0     0.00     0.00     0.00                -SUCCESS PUBLI         0     0.00     0.00     0.00
      -DELETE NAME                  0     0.00     0.00     0.00                -NO CACHE              0     0.00     0.00     0.00
      -READ CASTOUT STATS           0     0.00     0.00     0.00                -OVERWRITE AUT         0     0.00     0.00     0.00
                                                                                -OVERWRITE ENT         0     0.00     0.00     0.00
----GLOBAL DDF ACTIVITY------ -----QTY  /MINUTE  /THREAD  /COMMIT               -ENTRY NOT ADD         0     0.00     0.00     0.00
DBATS QUEUED(AT MAX)                0     0.00     0.00     0.00
CONVERSAT. DEALLOC(AT MAX)          0     0.00     0.00     0.00
INACTIVE DBATS-CURRENT              0
              -MAX                  0
ACTIVE DBATS  -CURRENT              0
              -MAX                  0
TOTAL DBATS   -MAX                  0
COLD START CONNECTIONS              0     0.00     0.00     0.00
WARM START CONNECTIONS              0     0.00     0.00     0.00
RESYNC ATTEMPTS                     0     0.00     0.00     0.00
RESYNC SUCCESSES                    0     0.00     0.00     0.00
```

Figure 26. Statistics Long Report—Page 4

For a complete description of each section, see

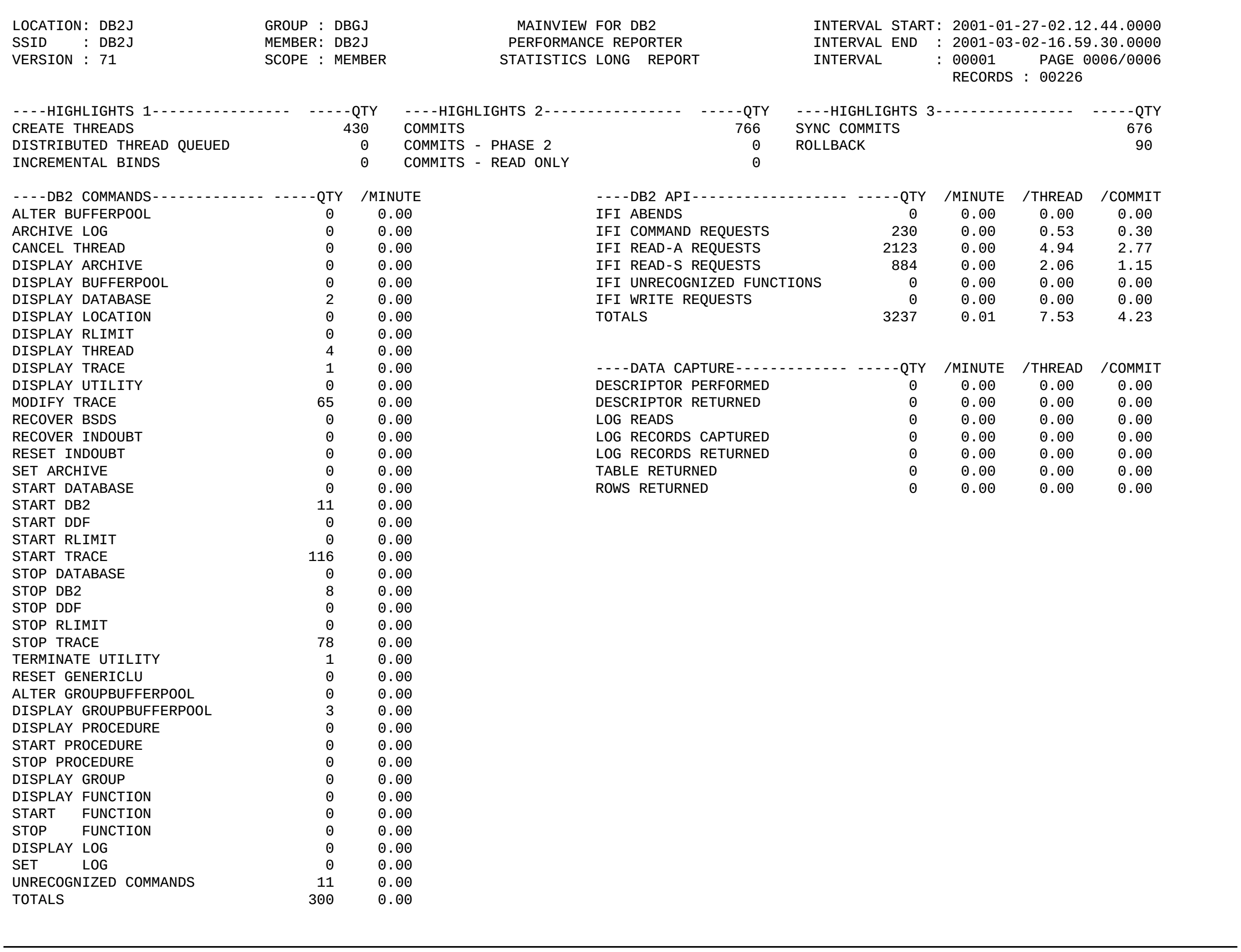
- “GLOBAL BUFFERPOOL” on page 290
- “GLOBAL LOCKING ACTIVITY” on page 296
- “GLOBAL DDF ACTIVITY” on page 301
- “AUTHORIZATION MANAGEMENT” on page 303

```
LOCATION: DB2J              GROUP : DBGJ                 MAINVIEW FOR DB2              INTERVAL START: 2001-01-27-02.12.44.0000
SSID    : DB2J              MEMBER: DB2J                PERFORMANCE REPORTER           INTERVAL END  : 2001-03-02-16.59.30.0000
VERSION : 71                SCOPE : MEMBER             STATISTICS LONG  REPORT         INTERVAL      : 00001     PAGE 0005/0006
                                                                                                      RECORDS : 00226

----HIGHLIGHTS 1----------------  -----QTY   ----HIGHLIGHTS 2----------------  -----QTY   ----HIGHLIGHTS 3----------------  -----QTY
CREATE THREADS                         430    COMMITS                                766    SYNC COMMITS                           676
DISTRIBUTED THREAD QUEUED                0    COMMITS - PHASE 2                        0    ROLLBACK                                90
INCREMENTAL BINDS                        0    COMMITS - READ ONLY                      0

----BP GENERAL--------------- -----QTY  /MINUTE  /THREAD  /COMMIT  ----BP WRITE OPERATIONS------ -----QTY  /MINUTE  /THREAD  /COMMIT
CURRENT ACTIVE BUFFERS            160                              PAGE UPDATES                    50042     0.11   116.38    65.33
VP BUFFER POOL FULL                 0     0.00     0.00     0.00   PAGES WRITTEN                    9369     0.02    21.79    12.23
SUCCESSFUL OPEN                  2860     0.01     6.65     3.73   BUFF UPDATES/PAGES WRITTEN          5     0.00     0.01     0.01
VP BUFFERS ALLOCATED             3700                              SYNC WRITES                       252     0.00     0.59     0.33
HP BUFFERS ALLOCATED                0                              ASYNC WRITE IO                    869     0.00     2.02     1.13
EXPANDED STORAGE HP BUFFERS         0                              ASYNC WRITES + SYNC WRITES       1121     0.00     2.61     1.46
MIGRATED DS ENCOUNTERED            84     0.00     0.20     0.11   HORIZONTAL DEF. WR REACHED          0     0.00     0.00     0.00
RECALL TIMEOUTS                     0     0.00     0.00     0.00   VERTICAL DEFER WR REACHED          68     0.00     0.16     0.09
HP EXP/CONTRACTION                  0     0.00     0.00     0.00   DM CRITICAL REACHED                 0     0.00     0.00     0.00
VP EXP/CONTRACTION                  0     0.00     0.00     0.00   NO WRITE ENGINE                     0     0.00     0.00     0.00
EXPAND SOS FAIL                     0     0.00     0.00     0.00   PAGES SYNC VP->HP                   0     0.00     0.00     0.00
HWM PREFETCH IO STREAMS            12                              PAGES ASYNC VP->HP                  0     0.00     0.00     0.00
PRFETCH IO STREAMS REDUCED          0                              PAGES WRITE FAIL VP->HP             0     0.00     0.00     0.00
REQUESTS FOR PARALLELISM           29     0.00     0.07     0.04   DATA MOVER ASYNC VP->HP             0     0.00     0.00     0.00
PARALLEL REDUCTION-NO BUFF          1     0.00     0.00     0.00   DATA MV ASYN FAIL VP->HP            0     0.00     0.00     0.00
PREFETCH QTY CUT TO 1/2           133     0.00     0.31     0.17   PAGEINS FOR WRITE IO               79     0.00     0.18     0.10
PREFETCH QTY CUT TO 1/4             0     0.00     0.00     0.00


----BP READ OPERATIONS------- -----QTY  /MINUTE  /THREAD  /COMMIT  ----BP SORT/MERGE------------ -----QTY  /MINUTE  /THREAD  /COMMIT
GETPAGES                     6511.14K    14.94   15.14K    8.50K   MAX WORKFILE IN MERGE               6
GETPAGE SEQ REQ              6392.70K    14.67   14.87K    8.35K   NUMBER MERGE PASSES                 7     0.00     0.02     0.01
GETPAGE RANDOM REQ            118.44K     0.27   275.44   154.62   MERGE PASSES/INSUFF BUFFER          0     0.00     0.00     0.00
SYNC READ I/O                   15545     0.04    36.15    20.29   WORKFILES REJECTED LOW BUFF         0     0.00     0.00     0.00
SYNC READ IO SEQ REQ             3067     0.01     7.13     4.00   TOTAL WORKFILES IN MERGE           16     0.00     0.04     0.02
SYNC READ IO RANDOM REQ         12478     0.03    29.02    16.29   WKFILES NOT CREATED-NO BUF          0     0.00     0.00     0.00
GETPAGE/SYNC READ RANDOM            9     0.00     0.02     0.01   PREFETCH DISABLED-WK FILES          0     0.00     0.00     0.00
SEQ PREFETCH REQ                23732     0.05    55.19    30.98   PAGES FOR DESTRUCTIVE READ       8280     0.02    19.26    10.81
SEQ PREFETCH READ IO            23318     0.05    54.23    30.44   DEQUE FROM VDWQ DISTRUCT RD      3853     0.01     8.96     5.03
SEQ PREFETCH PAGES READ       670.54K     1.54    1.56K   875.38
SEQ PREFETCH PAGES/READ            28     0.00     0.07     0.04
LIST PREFETCH REQUESTS              0     0.00     0.00     0.00
LIST PREFETCH READ IO               0     0.00     0.00     0.00
LIST PREFETCH PAGES READ            0     0.00     0.00     0.00
LIST PREFETCH PAGES/READ          N/C
DYNAMIC PREFETCH REQUESTS          78     0.00     0.18     0.10
DYNAMIC PREFETCH READ IO           30     0.00     0.07     0.04
DYNAMIC PREFETCH PAGES READ       662     0.00     1.54     0.86
DYNAMIC PREFETCH PAGES/READ        22     0.00     0.05     0.03
PF DISABLED - NO BUFFER             0     0.00     0.00     0.00
PREFETCH DISABLED-NO ENGINE         0     0.00     0.00     0.00
MVPG PAGES SYNC HP->VP              0     0.00     0.00     0.00
MVPG PAGES ASYNC HP->VP             0     0.00     0.00     0.00
HP->VP MVPG FAIL                    0     0.00     0.00     0.00
DATA MOVER ASYNC HP->VP             0     0.00     0.00     0.00
DATA MOV ASYNC FAIL HP->VP          0     0.00     0.00     0.00
PAGEINS FOR READ IO             15711     0.04    36.54    20.51
BPOOL HIT RATIO-ALL(%)             89
BPOOL HIT RATIO-RANDOM(%)          89
```

Figure 27. Statistics Long Report—Page 5

For a complete description of each section, see

- “BP GENERAL” on page 305
- “BP READ OPERATIONS” on page 309
- “BP WRITE OPERATIONS” on page 315
- “BP SORT/MERGE” on page 319

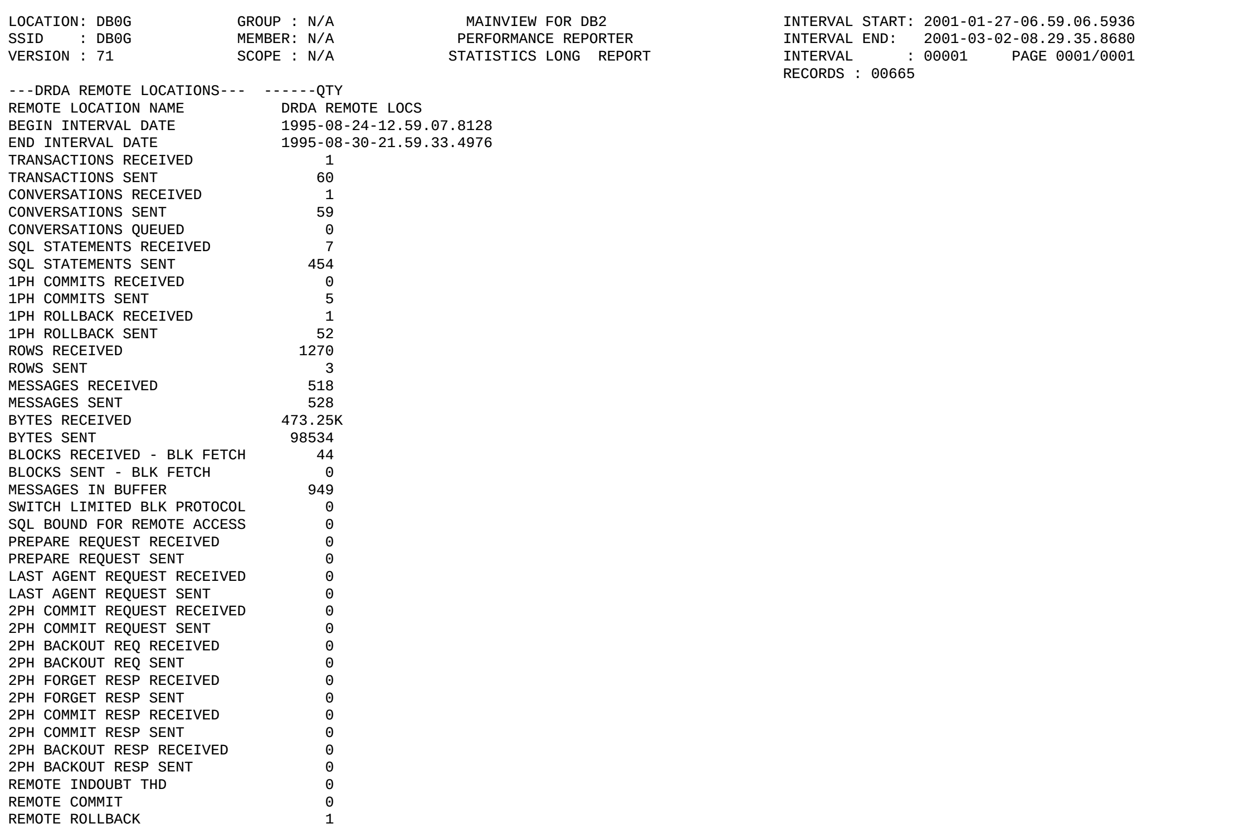


```
LOCATION: DB2J               GROUP : DBGJ                 MAINVIEW FOR DB2                  INTERVAL START: 2001-01-27-02.12.44.0000
SSID    : DB2J               MEMBER: DB2J                PERFORMANCE REPORTER               INTERVAL END  : 2001-03-02-16.59.30.0000
VERSION : 71                 SCOPE : MEMBER             STATISTICS LONG  REPORT             INTERVAL      : 00001     PAGE 0006/0006
                                                                                                            RECORDS : 00226

----HIGHLIGHTS 1----------------  -----QTY   ----HIGHLIGHTS 2----------------  -----QTY   ----HIGHLIGHTS 3----------------  -----QTY
CREATE THREADS                        430    COMMITS                                766   SYNC COMMITS                           676
DISTRIBUTED THREAD QUEUED               0    COMMITS - PHASE 2                        0   ROLLBACK                                90
INCREMENTAL BINDS                       0    COMMITS - READ ONLY                      0

----DB2 COMMANDS------------- -----QTY  /MINUTE                    ----DB2 API------------------ -----QTY  /MINUTE  /THREAD  /COMMIT
ALTER BUFFERPOOL                    0     0.00                     IFI ABENDS                          0     0.00     0.00     0.00
ARCHIVE LOG                         0     0.00                     IFI COMMAND REQUESTS              230     0.00     0.53     0.30
CANCEL THREAD                       0     0.00                     IFI READ-A REQUESTS              2123     0.00     4.94     2.77
DISPLAY ARCHIVE                     0     0.00                     IFI READ-S REQUESTS               884     0.00     2.06     1.15
DISPLAY BUFFERPOOL                  0     0.00                     IFI UNRECOGNIZED FUNCTIONS          0     0.00     0.00     0.00
DISPLAY DATABASE                    2     0.00                     IFI WRITE REQUESTS                  0     0.00     0.00     0.00
DISPLAY LOCATION                    0     0.00                     TOTALS                           3237     0.01     7.53     4.23
DISPLAY RLIMIT                      0     0.00
DISPLAY THREAD                      4     0.00
DISPLAY TRACE                       1     0.00                     ----DATA CAPTURE------------- -----QTY  /MINUTE  /THREAD  /COMMIT
DISPLAY UTILITY                     0     0.00                     DESCRIPTOR PERFORMED                0     0.00     0.00     0.00
MODIFY TRACE                       65     0.00                     DESCRIPTOR RETURNED                 0     0.00     0.00     0.00
RECOVER BSDS                        0     0.00                     LOG READS                           0     0.00     0.00     0.00
RECOVER INDOUBT                     0     0.00                     LOG RECORDS CAPTURED                0     0.00     0.00     0.00
RESET INDOUBT                       0     0.00                     LOG RECORDS RETURNED                0     0.00     0.00     0.00
SET ARCHIVE                         0     0.00                     TABLE RETURNED                      0     0.00     0.00     0.00
START DATABASE                      0     0.00                     ROWS RETURNED                       0     0.00     0.00     0.00
START DB2                          11     0.00
START DDF                           0     0.00
START RLIMIT                        0     0.00
START TRACE                       116     0.00
STOP DATABASE                       0     0.00
STOP DB2                            8     0.00
STOP DDF                            0     0.00
STOP RLIMIT                         0     0.00
STOP TRACE                         78     0.00
TERMINATE UTILITY                   1     0.00
RESET GENERICLU                     0     0.00
ALTER GROUPBUFFERPOOL               0     0.00
DISPLAY GROUPBUFFERPOOL             3     0.00
DISPLAY PROCEDURE                   0     0.00
START PROCEDURE                     0     0.00
STOP PROCEDURE                      0     0.00
DISPLAY GROUP                       0     0.00
DISPLAY FUNCTION                    0     0.00
START   FUNCTION                    0     0.00
STOP    FUNCTION                    0     0.00
DISPLAY LOG                         0     0.00
SET     LOG                         0     0.00
UNRECOGNIZED COMMANDS              11     0.00
TOTALS                            300     0.00
```

Figure 28.  Statistics Long Report—Page 6

For a complete description of each section, see

- “DB2 COMMANDS” on page 321
- “DB2 API” on page 322
- “DATA CAPTURE” on page 323



```
LOCATION: DB0G            GROUP : N/A              MAINVIEW FOR DB2                INTERVAL START: 2001-01-27-06.59.06.5936
SSID    : DB0G            MEMBER: N/A             PERFORMANCE REPORTER             INTERVAL END:   2001-03-02-08.29.35.8680
VERSION : 71              SCOPE : N/A            STATISTICS LONG  REPORT           INTERVAL      : 00001     PAGE 0001/0001
                                                                                   RECORDS : 00665
---DRDA REMOTE LOCATIONS---  ------QTY
REMOTE LOCATION NAME           DRDA REMOTE LOCS
BEGIN INTERVAL DATE            1995-08-24-12.59.07.8128
END INTERVAL DATE              1995-08-30-21.59.33.4976
TRANSACTIONS RECEIVED               1
TRANSACTIONS SENT                  60
CONVERSATIONS RECEIVED              1
CONVERSATIONS SENT                 59
CONVERSATIONS QUEUED                0
SQL STATEMENTS RECEIVED             7
SQL STATEMENTS SENT               454
1PH COMMITS RECEIVED                0
1PH COMMITS SENT                    5
1PH ROLLBACK RECEIVED               1
1PH ROLLBACK SENT                  52
ROWS RECEIVED                    1270
ROWS SENT                           3
MESSAGES RECEIVED                 518
MESSAGES SENT                     528
BYTES RECEIVED                 473.25K
BYTES SENT                      98534
BLOCKS RECEIVED - BLK FETCH        44
BLOCKS SENT - BLK FETCH             0
MESSAGES IN BUFFER                949
SWITCH LIMITED BLK PROTOCOL         0
SQL BOUND FOR REMOTE ACCESS         0
PREPARE REQUEST RECEIVED            0
PREPARE REQUEST SENT                0
LAST AGENT REQUEST RECEIVED         0
LAST AGENT REQUEST SENT             0
2PH COMMIT REQUEST RECEIVED         0
2PH COMMIT REQUEST SENT             0
2PH BACKOUT REQ RECEIVED            0
2PH BACKOUT REQ SENT                0
2PH FORGET RESP RECEIVED            0
2PH FORGET RESP SENT                0
2PH COMMIT RESP RECEIVED            0
2PH COMMIT RESP SENT                0
2PH BACKOUT RESP RECEIVED           0
2PH BACKOUT RESP SENT               0
REMOTE INDOUBT THD                  0
REMOTE COMMIT                       0
REMOTE ROLLBACK                     1
```

Figure 29. Statistics Long Report—DDF

For a complete description of each section, see

- “DRDA REMOTE LOCATIONS” on page 324

## Statistics Long Report—Page 1

The fields in this section are organized in the following categories.

For all count fields (not snapshot or high-water mark fields), four values are shown:

- The quantity counted in this reporting interval
- The value per minute, calculated by dividing the count by the interval elapsed time
- The value per thread, calculated by dividing the count by the total threads
- The value per commit, calculated by dividing the count by the number of commits

### HIGHLIGHTS

CREATE THREADS
Number of create thread requests (Q3STCTHD).

This reflects the number of threads created (not including distributed database access threads).

A thread is created at first execution of an SQL statement (assuming that no pre-existing thread for the same authorization ID and plan name exists).

DISTRIBUTED THREADS QUEUED
Number of times a DBAT thread was queued (QDSTQDBT).

**Tuning Tip:** This counter indicates that a requester thread was queued because the serving system did not allow enough active remote threads. The tuning parameter is MAXDBAT in DSNZPARM.

If this situation occurs often, increase MAXDBAT in DSNZPARM on installation panel DSNTIPE; reassemble and restart DB2.

INCREMENTAL BINDS
Number of incremental BINDs performed (QXINCRB).

This counter is incremented every time a plan is run that had been bound with the VALIDATE(RUN) option.

**Tuning Tip:** It is generally undesirable to bind a plan with the VALIDATE(RUN) option since all SQL statements must be rechecked for syntax, authority, and access path every time the plan is executed. VALIDATE(RUN) is required if the program is going to CREATE TABLES (for example, work tables) during the execution, or if testing is required on a piece of code for which the objects do not yet exist. Otherwise, VALIDATE(RUN) should be avoided, as the cost is nearly that of dynamic SQL.

COMMITS
Total number of commits and rollbacks listed below (Q3STCOMM+Q3STRDON+Q3STSYNC+Q3STABRT).

COMMITS - PHASE 2

Number of successful phase 2 requests (Q3STCOMM).

This is a counter of successful phase 2 commits from IMS or CICS transactions.

This field is not incremented for distributed two-phase commits or single unit of work tasks (for example, TSO, batch).

COMMITS - READ ONLY

Number of read-only commits (Q3STRDON).

This situation occurs only in CICS and IMS when execution of a program has not updated a DB2 resource. When this occurs DB2 simply increments the read-only counter, performs both phases of the two-phase commit process and records that the job was read-only. This count does not include CICS synchronous commits.

**Tuning Tip:** This information is useful in determining the read/write ratio of a CICS or IMS system.

SYNC COMMITS

Number of successful single phase commits (Q3STSYNC).

This is a count of all synchronous commits issued by TSO, batch, CAF, and utility programs. CICS applications use both synchronous commits and two-phase commits. IMS uses only two-phase commits.

ROLLBACK

Number of successful rollbacks (Q3STABRT).

This is a count of the number of rollbacks taken by the system to back out a unit of recovery.

Rollbacks occur because of a program abend, application rollback request, deadlock or timeout situation, a -CANCEL THREAD command, or some resource shortage. If it is a resource shortage, a -904 return code message is displayed on the MSTR job log.

## SQL DML

This section shows the number of each of these SQL DML statements:

- SELECT
- INSERT
- UPDATE
- DELETE
- PREPARE
- DESCRIBE
- DESCRIBE TABLE
- OPEN
- FETCH
- CLOSE

The total number of SQL DML statements processed is shown at the bottom of the column.

## SQL DDL

This section shows the number of each of these SQL DDL statements:

- ALTER DATABASE
- ALTER STORAGE GROUP
- ALTER INDEX
- ALTER TABLE
- ALTER TABLESPACE
- ALTER PROCEDURE (DB2 6.1 and later)
- ALTER FUNCTION (DB2 6.1 and later)
- CREATE ALIAS
- CREATE DATABASE
- CREATE INDEX
- CREATE STORAGE GROUP
- CREATE SYNONYM
- CREATE TABLE
- CREATE TABLESPACE
- CREATE VIEW
- CREATE TEMP TABLE (DB2 5.1 and later)
- CREATE AUX TABLE (DB2 6.1 and later)
- CREATE PROCEDURE (DB2 6.1 and later)
- CREATE FUNCTION (DB2 6.1 and later)
- CREATE TRIGGER (DB2 6.1 and later)
- CREATE DISTINCT TYPE (DB2 6.1 and later)
- DROP ALIAS
- DROP DATABASE
- DROP INDEX
- DROP STORAGE GROUP
- DROP SYNONYM
- DROP TABLE
- DROP TABLESPACE
- DROP VIEW
- DROP PACKAGE (DB2 6.1 and later)
- DROP PROCEDURE (DB2 6.1 and later)
- DROP FUNCTION (DB2 6.1 and later)
- DROP TRIGGER (DB2 6.1 and later)
- DROP DISTINCT TYPE (DB2 6.1 and later)

The total number of SQL DDL statements processed is shown at the bottom of the column.

## SQL DCL

LOCK TABLE

Number of LOCK TABLE statements (QXLOCK).

This counter is incremented every time a LOCK TABLE statement is executed (either SHARE or EXCLUSIVE).

GRANT

Number of GRANT statements (QXGRANT).

This counter is incremented every time a GRANT statement is executed.

**Tuning Tip:** This field is useful for the auditor who wishes to monitor the granting of authority for a specific time period. Other audit traces can be activated to capture which authorities were granted. The catalog can also be queried.

REVOKE

Number of REVOKE statements (QXREVOK).

This counter is incremented every time a REVOKE statement is executed.

**Tuning Tip:** This field is useful for the auditor who wishes to monitor the revocation of authority for a specific time period. Other audit traces can be activated to capture which authorities were revoked.

SET CURRENT SQLID

Number of SET CURRENT SQLID statements (QXSETSQL).

This counter is incremented every time a SET CURRENT SQLID statement is executed.

SET HOST VARIABLE

Number of SET HOST VARIABLE statements (QXSETHV).

This counter is incremented every time a SET HOST VARIABLE statement is executed.

SET CURRENT DEGREE

Number of SET CURRENT DEGREE statements executed (QXSETCDG).

This counter is incremented every time an application program issues a SET CURRENT DEGREE register to enable or disable parallel processing for dynamic SQL.

SET CURRENT RULES

Number of SET CURRENT RULES statements executed (QXSETCRL).

This counter is incremented every time an application program issues a SET CURRENT RULES register to change syntax parsing from SQL rules to ANSI/SQL processing. (DB2 4.1 and later)

**Tuning Tip:** This number may be useful in determining the amount of non-DB2 traffic being executed during this interval period.

SET CURRENT PATH

Number of SET CURRENT PATH statements executed (QXSETPTH).

(DB2 6.1 and later)

CONNECT TYPE 1

Number of CONNECT TYPE 1 statements executed (QXCON1).

This is a count of the number of CONNECT TYPE 1 statements processed.

Type 1 SQL CONNECT statements allow one updateable site in the connection. The type of Connect statement is specified as a precompiler parameter CONNECT(1) or CONNECT(2).

CONNECT TYPE 2

Number of CONNECT TYPE 2 statements executed (QXCON2).

This is a count of the number of CONNECT TYPE 2 statements processed.

CONNECT TYPE 2 SQL statements allow multiple updateable sites in the distributed connection. This type of Connect statement is specified as a precompiler parameter CONNECT(1) or CONNECT(2).

SET CONNECTION

Number of SET CONNECTION statements executed (QXSETCON).

This counter is incremented every time an application program issues a SET CONNECT SQL verb. This verb is required to perform multi-site updates.

RELEASE

Number of RELEASE statements executed (QXREL).

This counter is incremented every time an SQL RELEASE statement is issued terminating a remote conversation.

**Tuning Tip:** This number is useful in determining the activity of network conversations during the interval period.

CALL

Number of CALL statements executed (QXCALL).

This counter is incremented every time an application program issues a CALL statement to execute a DB2 stored procedure (DB2 4.1 and later).

ASSOCIATE LOCATOR

Number of ASSOCIATE LOCATOR statements executed (QXALOCL).

These statements get the result set locator value for each result set returned by a stored procedure (DB2 5.1 and later).

HOLD LOCATOR

Number of HOLD LOCATOR statements executed (QXHOLDL).

(DB2 6.1 and later)

FREE LOCATOR

Number of FREE LOCATOR statements executed (QXFREEL).

(DB2 6.1 and later)

ALLOCATE CURSOR
Number of ALLOCATE CURSOR statements executed (QXALOCC).

These statements define a cursor and associate it with the result set locator variable (DB2 5.1 and later).

TOTALS
Total number of SQL DCL statements processed.

This section also shows the number of each of the following SQL statements:

- COMMENT ON
- RENAME TABLE (DB2 5.1 and later)
- LABEL ON

## OPTIMIZATION

These fields apply only to DB2 5.1 and later.

PREPARE - SATISFIED
Number of times DB2 satisfied a PREPARE request by making a copy of a statement in the prepared statement cache (QXSTFND).

PREPARE - FAILED SEARCH
Number of times DB2 searched the prepared statement cache but could not find a suitable prepared statement (QXSTNFND).

PREPARE - IMPLICIT
Number of times DB2 did an implicit PREPARE for a statement bound with KEEPDYNAMIC(YES) because the prepared statement cache did not contain a valid copy of the prepared statement (QXSTIPRP).

PREPARE - FROM CACHE
Number of times DB2 did not prepare a statement bound with KEEPDYNAMIC(YES) because the prepared statement cache contained a valid copy of the prepared statement (QXSTNPRP).

PREPARE - DISCARDED MAX
Number of times DB2 discarded a prepared statement from the prepared statement cache because the number of prepared statements in the cache exceeded the value of subsystem parameter MAXKEEPD (QXSTDEXP).

PREPARE - DROP/ALTER/REVOKE
Number of times DB2 discarded a prepared statement from the prepared statement cache because a program executed a DROP, ALTER, or REVOKE statement against a dependent object (QXSTDINV).

## RID LIST

RID CURRENT BLOCKS ALLOCATE
Current number of RID blocks in use (QISTRCUR).

This is a snapshot of the number of RID blocks in use.

**Tuning Tip:** If this number is at the RID pool size, check for failures. Increase the DSNZPARM parameter MAXRBLK in DSNZPARM to the maximum possible to support list prefetch and multiple index access paths.

RID HIGH BLOCKS ALLOCATED
Maximum number of RID blocks in use (QISTRHIG).

This is the maximum number of RID blocks in use at any one time.

**Tuning Tip:** If this number is at the RID pool size, check for failures. Increase the DSNZPARM parameter MAXRBLK in DSNZPARM to the maximum possible to support list prefetch and multiple index access paths.

RID TERM OVER RDS LIMIT
Number of RID pool failures—RID limit (QISTRLLM).

This counter is incremented every time a RID pool failure occurred due to a single set of index entries occupying more than 50% of the RID pool.

**Tuning Tip:** This number should be as close to zero as possible since the query degrades to a table space scan if a failure occurs. Increase the size of the RID pool in DSNZPARM MAXRBLK on installation panel DSNTIPC.

RID TERM TOO MANY CONCURR
Number of RID pool failures—too many concurrent users (QISTRMAX).

This counter is incremented every time a RID pool failure occurred because there were too many concurrent users.

**Tuning Tip:** Increase the DSNZPARM parameter MAXRBLK in DSNZPARM to the maximum possible to support list prefetch and multiple index access paths.

RID TERM - NO STORAGE
Number of RID pool failures—storage exceeded (QISTRSTG).

This counter is incremented every time a RID pool failure occurred due to a lack of storage.

**Tuning Tip:** Increase the DSNZPARM parameter MAXRBLK in DSNZPARM to the maximum possible to support list prefetch and multiple index access paths.

RID TERM - OVER DM LIMIT
Number of RID pool failures—data manager limit (QISTRPLM).

This counter is incremented when a very large RID list is encountered. The maximum RID list is 16 million RIDS.

**Tuning Tip:** To avoid this, change the SQL statement so a different access path is chosen. If a failure does occur, the query is run as a table space scan.

## OPEN/CLOSE ACTIVITY

SLOW CLOSE DATASETS

Current number of deferred close data sets (QTSLWDD).

This is a snapshot count of the number of data sets that are not in use but remain in deferred close status because DB2 has not reached a threshold which would involve physical close activity.

DB2 attempts to avoid physical VSAM OPENS and CLOSES. It places the unused data sets on the deferred close queue to avoid the REOPEN cost when next used. The number of *in-use* data sets is the difference between the currently open and currently deferred closed data sets.

DATASETS CURRENTLY OPEN

Number of data sets currently open (QTDSOPN).

This snapshot count reflects all open table space and index space data sets at the time the statistics record was cut.

DB2 DSNZPARM DSMAX specifies the total storage which can be allocated for open data sets. This number cannot exceed 10,000.

MAX SLOW CLOSE DATASETS

Maximum number of page sets on deferred close queue (QTMAXPB).

This represents the high-water mark of data sets that are not in use and not physically closed.

**Tuning Tip:** DB2 goes through pseudo-close processing because the expense of VSAM OPEN and CLOSE is quite high. But when the DSMAX or 10,000 data set limit is reached, CLOSE(YES), then those with CLOSE(NO), using a least recently used (LRU) algorithm. This value can be greater than DSMAX.

MAX D.S. OPEN CONCURRENTLY

Maximum number of open data sets (QTMAXDS).

This number reflects the high-water mark of all open data sets.

When DB2 reaches 99% of the DSMAX value, it starts physically closing page sets, first those with CLOSE(YES), then those with CLOSE(NO), using a least recently used (LRU) algorithm. This value can be greater than DSMAX.

D.S CLOSED (THRESHOLD)

Number of unused data sets closed due to deferred close (QTDSDRN).

This counter is incremented when a data set has not been recently used and DB2 has reached 99% of DSMAX. DB2 closes these data sets.

CLOSE YES data sets are chosen before CLOSE NO data sets.

D.S. CONVERTED - RW TO RO

Number of data sets switched from R/W to R/O (QTPCCT).

This counter is incremented when a data set has not been updated for a specified number of checkpoints (ZPARM PCLOSEN) or a specified number of minutes (ZPARM PCLOSET).

This feature speeds recovery by posting a SYSLGRNX or SYSLGRNG close entry. This means that fewer log records are necessary to accomplish a recovery since DB2 knows that no updates have been applied while the data set is marked read-only. The data set is switched back to R/W automatically at the first update SQL statement.

REOPEN FROM SLOW CLOSE

Number of times a deferred close data set is reused (QTREOPN).

This counter is incremented each time a data set is accessed that was on the deferred close queue. Although the data set was not in use, it was not physically closed, so did not have to be reopened.

**Tuning Tip:** The higher this number is, the better the system use of deferred close.

## Statistics Long Report—Page 2

The fields in this section are organized in the following categories. For the highlights, see page 254.

### EDM POOL

PAGES IN EDM POOL
Number of pages in the EDM pool (QISEPAGE).

This is the size, in 4K pages, of the EDM pool. This is a snapshot value.

This is set in DSNZPARM EDMPOOL on the installation panel DSNTIPC.

FREE PAGES
Number of free pages (QISEFREE).

This counter is a snapshot of how many pages in the EDM pool are not being used by any DBD, SKCT, SKPT, CT or PT.

**Tuning Tip:** A rough rule of thumb is that in a production system in which the workload is predictable, at least 15% of the EDM pool pages should be free to account for an unanticipated load of a large DBD. This can vary, however, since many shops run more plans than could practically be kept in memory.

EDM POOL FULL FAILURES
Number of failures because EDM pool was full (QISEFAIL).

This counter is incremented when an operation cannot proceed because the EDM pool was full. This is an extremely undesirable situation in which all pages in the Environmental Descriptor Manager pool (EDM pool) are allocated and in use as database descriptors (DBDs), skeleton cursor tables (SKCTs, which are internal copies of plans), skeleton package tables (SKPTs, which are internal copies of packages), and working cursor tables (CTs), and working package tables (PTs). No other operation can be commenced until one or more pages are freed in this pool.

**Tuning Tip:** This number should be as close to zero as possible. If this situation occurs, consider increasing the number of pages in the EDM pool in DSNZPARM, on installation panel DSNTIPC.

CT PAGES USED
Number of pages used for cursor table sections (QISECT).

This is a snapshot of the number of pages allocated to cursor table sections (CTs).

**Tuning Tip:** The ideal goal is to keep as many SKCTs, SKPTs, and DBDs in the EDM pool as possible to avoid I/O to the DB2 Directory.

DBD PAGES USED

Number of pages used for DBDs (QISEDBD).

This counter is a snapshot of the number of pages currently occupied by database descriptors (DBDs).

**Tuning Tip:** This count is the most important count of EDM pool objects because of an inherent rule that DBDs must occupy contiguous storage. SKPTs, SKCTs, CTs, or PTs do not have this restriction. This may result in EDM pool fragmentation. Ways of avoiding this problem include keeping DBD sizes to a reasonable size by not allowing more than 100 objects per DB2 database. If the number of DBD pages occupy more than half the EDM pool, consider increasing the size of the pool.

PKG TBL PAGES USED

Number of pages used for package tables (QISEKT).

This is a snapshot of the number of pages allocated to working package tables (PTs).

This number reflects the amount of storage consumed by working package tables.

SKCT PAGES USED

Number of pages used for skeleton cursor tables (QISESKCT).

This is a snapshot of the number of EDM pool pages allocated to skeleton cursor tables (SKCTs).

Skeleton cursor tables are the DB2 plan information which is read from directory table SCT01.

SKPT PAGES USED

Number of pages used for skeleton package tables (QISESKPT).

This is a snapshot of the number of pages allocated to skeleton package tables (SKPTs).

This value reflects the amount of storage used by skeleton package tables.

CT REQUESTS

Number of requests for cursor table sections (QISECTG).

**Tuning Tip:** This number is incremented every time DB2 needs a new 4K cursor table section (CT) of the skeleton cursor table (SKCT). The request can be resolved by looking for a copy in the EDM pool; if a copy does not exist, a read operation must be made to the directory table SCT01.

This number reflects how many sections were requested during a specific interval period.

CT NOT IN EDM POOL

Number of times CT sections not found in the EDM pool (QISECTL).

This counter is incremented every time a request for a cursor table section (CT) is made for which a copy in an SKCT does not exist in the EDM pool.

Every time this occurs, an I/O is scheduled to the SCT01 directory table. It is desirable to reduce the I/O on these very volatile directory page sets.

CT REQUESTS/CT NOT IN POOL

Ratio of the number of requests for cursor table sections to the number of times CT sections not found in the EDM pool (QISECTG/QISECTL).

DBD REQUESTS

Number of requests for DBDs (QISEDBDG).

This number reflects the total number of requests for database descriptors (DBDs).

Requests for DBDs are honored by either finding a working copy previously loaded into the EDM pool or by reading directory table DBD01.

DBD NOT IN EDM POOL

Number of times DBD not found in the EDM pool (QISEDBDL).

This counter reflects the number of times the database descriptor for a table is not found in the EDM pool. It must be read from the catalog table DBD01 into contiguous storage in the EDM pool.

**Tuning Tip:** If possible, the DBDs can be preloaded into the EDM pool by issuing the -DISPLAY DATABASE(*) command to seed the EDM pool right after DB2 start. This may not account for all fragmentation which can occur because of age, but it can establish a baseline of how large the pool should be sized.

DBD REQ/DBD NOT IN POOL

Ratio of the number of requests for DBDs to the number of times DBD not found in the EDM pool (QISEDBDG/QISEDBDL).

PKG TBL REQUESTS

Number of requests for package table sections (QISEKTG).

This counter is incremented every time a package section (PT) is requested.

**Tuning Tip:** Ideally, an SKPT (skeleton package table) with this PT will already be in the EDM pool. If not, the SKPT must be read from directory table SPT01.

PKG TBL NOT IN EDM POOL

Number of times package table sections not found in EDM pool (QISEKTL).

This counter is incremented every time a request for a skeleton package table section is made on behalf of a calling thread but the SKPT was not found in the EDM pool.

**Tuning Tip:** Ideally, an SKPT (skeleton package table) will already be in the EDM pool. If not, it must be read from directory table SPT01.

PKG TBL REQ/PKG NOT IN POOL

Ratio of the number of requests for package table sections to the number of times package table sections not found in EDM pool (QISEKTG/QISEKTL).

PREP STMT CACHE-INSERTS
Number of inserts into the dynamic statement cache (QISEDSI).

(DB2 6.1 and later)

PREP STMT CACHE-REQUESTS
Number of requests for the dynamic statement cache (QISEDSG).

(DB2 6.1 and later)

PREP STMT CACHE-PAGES USED
Number of pages used for the dynamic statement cache (QISEDSC).

(DB2 6.1 and later)

EDM DATA SPACE-PAGES
Number of pages in the data space used by the EDM pool (QISEDPGE).

(DB2 6.1 and later)

EDM DATA SPACE-FREE PAGES
Number of free pages in the data space free chain (QISEDFRE).

(DB2 6.1 and later)

EDM DATA SPACE-FAILURES
Number of failures because the data space is full (QISEDFAL).

(DB2 6.1 and later)

## SYSTEM SERVICES

IDENTIFY REQUEST

Successful identify requests (Q3STIDEN).

This reports the number of threads that have gone through successful connection processing to the DB2 being observed. Connections can be from any supported environment, such as TSO, IMS, CICS, CAF (call attach), or a utility.

**Tuning Tip:** Watch for peak time spike periods. The DB2 DSNZPARM parameters CTHREAD, IDFORE, IDBACK, and other distributed parameters should be adjusted so the system can respond to the maximum load expected.

CREATE THREAD

Create thread requests (Q3STCTHD).

This reflects the number of threads created (not including distributed database access threads).

A thread is created at first execution of an SQL statement (assuming that no pre-existing thread for the same authorization ID and plan name exists).

CREATE THREAD THAT WAITED

Number of create thread requests queued (Q3STCTHW).

This field is a counter of how many times the maximum thread count (CTHREAD in DSNZPARM) was reached and a user had to wait to acquire an available thread. It does not include DBATs.

**Tuning Tip:** The general rule is to keep CTHREAD high enough to avoid thread queuing. However, if memory resources are tight (and the users are willing to live with the pain), you can limit CTHREAD to an arbitrary number. This controls memory allocation for thread creation at the expense of user wait time and some overhead in keeping track of the queued threads for scheduling.

SIGNON

Successful signon events (Q3STSIGN).

This field has meaning only when applied to CICS and IMS. It represents the number of times an existing thread has gone through new user signon reusing an existing thread.

**Tuning Tip:** The cost of thread creation can be quite considerable in CICS transactions and IMS WFI BMP transactions. In general, it is desirable to reuse an existing thread rather than suffer the destruction and recreation of a new thread. High numbers generally indicate good thread reuse. Tuning factors which can impact this are GRANTing plans to PUBLIC (avoiding an authorization check) or utilizing the authorization cache by specifying a large enough CACHESIZE at bind time to keep as many user IDs in the EDM pool as possible.

TERMINATE

Successful terminate thread requests (Q3STTERM).

This number reflects thread termination as a result of a program releasing resources or of a thread dropping its access level from *thread active* back to signon and back to identify status. This count will be higher than the create thread count.

**Tuning Tip:** In online transactions, keep an eye on thread creations and terminations for similar transactions. Thread reuse may be achieved in CICS by raising the THRDS count in the CICS resource control table; in an IMS environment, a Wait-for-Input WFI) BMP can be a useful method of avoiding costly thread creations and terminations.

ROLLBACK

Successful rollbacks (Q3STABRT).

This is a count of the number of rollbacks taken by the system to back out a unit of recovery.

Rollbacks occur because of a program abend, application rollback request, deadlock or timeout situation, a -CANCEL THREAD command, or by some resource shortage. If it is a resource shortage, a -904 return code message is displayed on the MSTR job log.

PREPARE TO COMMIT (PHASE 1)

Successful prepare to commits (Q3STPREP).

This is a count of the number of *prepare to commit* requests for a two-phase commit unit of work, which includes CICS update and IMS transactions. The prepare to commit is the result of the end of phase one which causes log records to be externalized. This counter is appropriate only for two-phase commit operations.

The use of this field is to determine how much forced logging is occurring as a result of end of phase 1 processing. The number of prepares for phase 1 minus the successful commit phase 2 requests is the indoubt count. However, since some phase 2 operations do not complete in the same time interval, the number of prepares for phase 1 do not always agree with successful phase 2 requests.

COMMIT (PHASE 2)

Successful phase 2 requests (Q3STCOMM).

This is a counter of successful phase 2 commits from IMS or CICS transactions.

This field is not incremented for distributed two-phase commits or single unit of work tasks (for example, TSO, batch).

READ-ONLY COMMIT

Number of read-only commits (Q3STRDON).

This situation occurs only in CICS and IMS when execution of a program has not updated a DB2 resource. When this occurs, DB2 simply increments the read-only counter, performs both phases of the two-phase commit process, and records that the job was read-only. This count does not include CICS synchronous commits.

**Tuning Tip:** This information is useful in determining the read/write ratio of a CICS or IMS system.

SYNCH COMMIT (SINGLE PHASE)

Successful single-phase commits (Q3STSYNC).

This is a count of all synchronous commits issued by TSO, batch, CAF, and utility programs. CICS applications use both synchronous commits and two-phase commits. IMS uses only two-phase commits.

INDOUBT

Total number of indoubt units of recovery (Q3STINDT).

This is a count of indoubt threads, caused when a failure occurs after a successful prepare but before a successful commit. The failure can occur in the address space of the application, the transaction manager, DB2, or a distributed requester / server.

**Tuning Tip:** Most indoubt situations are resolved automatically when the participants are restarted. These are counted in *successful indoubt resolutions* (Q3STRIUR). However, in some cases, such as an operator cold starting one participant, manual resolution may be required.

INDOUBT RESOLVED

Successful indoubt resolutions (Q3STRIUR).

This is a count of successful indoubt thread resolutions. Indoubt situations arise when a failure occurs after a successful prepare but before a successful commit. The status of retained locks against resources cannot be resolved until the coordinator and all participants have been recovered/restarted. This resolution usually occurs automatically, as reflected in this count.

**Tuning Tip:** This field applies only to CICS and IMS transactions participating in two-phase commit or in distributed processing using DRDA level 2 processing. In some cases, such as an operator cold starting one participant, manual resolution may be required. Another counter shows the total number of indoubt threads (Q3STINDT).

END OF MEMORY

Abnormal allied memory end of memory (Q3STMEOM).

This counter reflects program abends due to end of memory situations (such as an MVS FORCE command or lack of region to successfully abend).

**Tuning Tip:** Non-zero numbers in this counter should be investigated to determine why FORCE commands were issued or an X78 abend occurred.

END OF TASK

Abnormal allied memory end of task (Q3STMEOT).

This field is a counter of those tasks which have abended while connected to DB2. The counts reflected should equal the number of Abnormal EOT messages in the MSTR job log.

**Tuning Tip:** This field is useful in determining how many abends occurred.

DB2 CHECKPOINT

Number of DB2 checkpoints (QWSDCKPT).

This counter describes the number of DB2 system checkpoints taken.

**Tuning Tip:** During checkpoint processing, important events are written to the log, including internal status information. This information is needed for recovery processing if DB2 abends or MVS terminates without proper shutdown.

## LOCKING ACTIVITY

TIMEOUT

Number of lock timeouts (QTXATIM).

This count is incremented every time a DB2 thread waits longer to get a page than the timeout interval. This is specified with DSNZPARM IRLMRWT on installation panel DSNTIPI. By default, it is 60 seconds. Utilities can be allowed to wait several multiples of IRLMRWT.

**Tuning Tip:** Lock timeouts are usually caused by an application failing to commit in time for the other thread to gain access to data on pages. Often the problem can be resolved by reducing the time between commits and putting updateable statements near to their commit logic. Every time a timeout occurs, DB2 writes the holder and suspender to the MSTR job log. Normally, this number should be as close to zero as possible.

DEADLOCK

Number of deadlocks (QTXADEA).

This count is incremented every time DB2 encounters a deadlock situation for which the IRLM must cancel a task involved in a *deadly embrace*.

**Tuning Tip:** Deadlocks are caused by threads requesting access to two resources which can never be resolved. DB2 chooses its victim by selecting the thread that has done the least number of updates. It records the deadlock in the MSTR address space job log. In well-tuned systems, this number should be low.

SUSPEND - LOCK CONFLICT

Number of suspends due to lock conflict (QTXASLOC).

This counter is incremented any time a thread has a conflicting lock request, such as an updater requesting exclusive access to a page being used by another thread.

**Tuning Tip:** In a multitasking system, suspensions occur in the normal course of the events. If applications are well-tuned, taking frequent commits and holding on to resources for the fewest possible instructions, suspensions can be minimized. If suspensions cause frequent timeouts, consider row-level locking (DB2 4.1 and later).

SUSPEND FOR LATCH

Number of suspends due to latch conflicts (QTXASLAT).

This number is incremented when a latch conflict exists between two DB2 threads or internal serialization processing.

**Tuning Tip:** Latches are generally of extremely short duration. Unless the time is a significant component of overall wait time, it is not a factor which should cause tuning problems.

SUSPEND OTHER REASONS

Number of suspends due to other conflicts (QTXASOTH).

This number is incremented when DB2 internal processes collide.

**Tuning Tip:** This number is not generally of significance in tuning. Unusually high numbers should be reported to IBM service.

LOCK ESCALATION TO SHARED

Number of lock escalations to shared mode (QTXALES).

This count is incremented every time the number of locks against a single table space exceeds the number set in DSNZPARM NUMLKTS on installation panel DSNTIPJ or in the LOCKMAX clause of the CREATE table space statement (DB2 4.1 and later).

**Tuning Tip:** This is not a normal situation unless using repeatable read (RR). If it occurs often, consider changing the LOCKSIZE or LOCKMAX to a higher value or consider binding the plan with cursor stability (CS) or uncommitted read (UR).

LOCK ESCALATION TO EXCLUSIVE

Number of lock escalations to exclusive mode (QTXALEX).

This counter is incremented every time the number of updateable locks against a single table space exceeds the DSNZPARM NUMLKTS on installation panel DSNTIPJ or in the LOCKMAX clause of the CREATE table space statement (DB2 4.1 and later). It occurs when the LOCKSIZE parameter is specified as ANY and DB2 has escalated the lock owner to an exclusive lock of the entire table.

**Tuning Tip:** This is an extremely undesirable situation, usually caused by leaving the LOCKSIZE(ANY) default. To resolve this situation, consider changing the parameter to LOCKSIZE(PAGE) or in special situations, LOCKSIZE(ROW) (DB2 4.1 and later). This causes the offending application to take the -904 unavailable resource error rather than cause general unavailability to the rest of the users. This is almost always caused by application failure to commit in a timely fashion and can be resolved by application code changes as well as by DBA action.

DRAIN REQUEST

Number of drain requests (QTXADRNO).

This counter is incremented each time a utility or command requests a serialization against a page set resource.

This number is of significance in determining the amount of utility and command activity that requests some serial access to a resource.

UNSUCCESSFUL DRAIN REQUEST

Number of unsuccessful drain requests (QTXADRUN).

This counter is incremented when a potential drainer (utility or command) cannot obtain use of a page set because the claim count has not dropped to zero within the utility timeout value set in IRLMWRT of DSNZPARM.

This number is of significance in determining the number of unsuccessful utility and command processes due to user activity in the resource.

CLAIM REQUEST

Number of claim requests (QTXACLNO).

This number is incremented every time a user executes an SQL statement that increments the use count of a table space, partition, or index space data set.

This number gives an overall level of SQL activity in this system.

UNSUCCESSFUL CLAIM REQUEST

Number of unsuccessful claim requests (QTXACLUN).

This number is incremented every time a user issues a request for a claim to an SQL resource but cannot acquire it, usually because a utility or command DRAIN is on the object being sought.

This number is of some significance in determining contention between SQL and other types of utilities or commands.

LOCK REQUEST

Number of lock requests (QTXALOCK).

This counter is incremented for each call to the IRLM lock manager to acquire a lock on a page or row or to acquire a claim or drain on a data set.

**Tuning Tip:** Each lock request is processed by the IRLM. Lock avoidance techniques should show reductions in overall counts and overhead, since latches execute totally within DB2.

UNLOCK REQUEST

Number of unlock requests (QTXAUNLK).

This count is incremented when the application has finished processing a page or row, or a claim or drain can be released.

This amount is significant as to cross memory processing and reflects the normal release of resources

LOCK QUERY REQUEST

Number of query requests (QTXAQRY).

This counter is incremented every time the IRLM gets a request to read data.

**Tuning Tip:** This information is useful in determining the read activity on a system but does not include lock avoidance techniques.

LOCK CHANGE REQUEST

Number of change requests (QTXACHG).

This counter is incremented every time the IRLM is asked to change a lock from one type to another (for example, from ‘S’ to ‘X’).

The information is useful in determining the overall volatility of the system.

LOCK OTHER IRLM REQUEST

Number of other IRLM requests (QTXAIRLM).

This counter is incremented every time the IRLM receives a lock request not included in the other counts.

## LOG ACTIVITY

LOG READS FROM BUFFER
Number of log reads from buffer (QJSTRBUF).

This counter is incremented when DB2 successfully finds the necessary data in a log buffer to perform a backout or recover operation.

This is the best situation if an application abends. The pages necessary to be *undone* are still in memory (the DB2 output log buffer) and backout should take place quickly.

LOG READS FROM ACTIVE
Number of log reads from active logs (QJSTRACT).

This counter is incremented when DB2 successfully finds the necessary data in the active log to perform a backout or recover operation. (The pages are no longer in the log output buffer.)

**Tuning Tip:** IBM’s general rule of thumb is that if storage is not a consideration, plan to have enough active logs on DASD to backout or recover for at least the prior 24 hours, without requiring access to the archive logs. If this is not possible, consider the longest unit of work to be run on the DB2 in question and have enough logs so in the event of an abend, calls to the archive log data set are avoided. It should be noted that backout takes approximately twice the time as the original update, since DB2 must write compensation records to the log while reading backwards from it.

LOG READS FROM ARCHIVE
Number of log reads from archive logs (QJSTRARH).

This counter is incremented when a backout or recover operation must go back to the archive logs to accomplish the task.

**Tuning Tip:** This is generally not desirable since tape or cartridge mounts must be performed to recover or back out data from the archive logs. If this number is consistently greater than zero, it is a good idea to increase the number of active logs to avoid unnecessary delays.

READ DELAY/UNAVAIL RESOURCE
Number of read accesses delayed due to resource unavailability (QJSTWUR).

This counter reflects the number of time a recovery or backout was delayed due to tape contention or not enough TCBs to process the number requested.

The -SET ARCHIVE command can increase the number of tape drives available to DB2.

READ ACCESS DELAY-TAPE CONT
Number of log read accesses delayed due to tape contention (QJSTTVC).

This counter is incremented when two or more backout or recover operations call for the same tape volume. DB2 holds and MVS enqueue until the volume is freed.

**Tuning Tip:** Ideally, this number should be zero. Consider using DFHSM or similar media to stage the archive data to DASD if this is a frequent occurrence. DB2 allows multiple tasks to share archive data on DASD as well as look-ahead tape mounts.

ACTIVE LOG WRITE REQUESTS

Number of calls to the log write routine (QJSTBFWR).

This counter is incremented every time log records need to be written.

The number does not represent the number of physical I/Os since DB2 attempts to chain CIs together before writing to the log.

ACTIVE LOG WRITE NOWAIT

Number of nowait log writes (QJSTWRNW).

This counter is incremented when log records are written directly to the log buffers without waiting for the write to the data set.

Unless an application program is seriously taking too many COMMITs, this is the normal asynchronous log processing which does not cost the application any wait time.

BSDS REQUESTS

Number of BSDS access requests (QJSTBSDS).

This counter increments every time the logging subsystem must access the bootstrap data set. DB2 maintains the RBA range for every active and archive log in the system.

**Tuning Tip:** This number will usually be non-zero, since the BSDS must be updated by DB2 every time an archive log process occurs (a copy of the BSDS is REPROed to the first file on the archive medium) and updates are made to the high and low RBA range. Over time, however, look for steadily increasing numbers in this counter. The bootstrap data set is the only key-sequenced VSAM data set (KSDS) used by DB2. As with any other KSDS, CI and CA splits can and do occur. These should be dealt with by REPROing the KSDS to a different file and/or redefining the BSDS as a larger data set.

ACTIVE LOG UNAVAIL BUFFER

Number of waits for unavailable log buffer (QJSTWTB).

This counter is incremented when DB2 must wait for an available output log buffer before it can externalize log records.

**Tuning Tip:** This counter should always be zero, and in all but the busiest DB2 environments will be zero, since it takes a large amount of updated pages to oversaturate the log buffers. Increase the OUTBUFF parameter in DSNZPARM on the DSNTIPL installation screen should this situation occur. All applications wait following a commit until log buffers become available.

ACTIVE LOG CIS CREATED

Number of active log output CIs created (QJSTBFFL).

This counter simply updates the number of 4K CIs created as part of log processing.

**Tuning Tip:** The number is useful in determining the heaviest logging periods in the system.

CONTROL INTERVALS OFFLOADED

Count of control intervals offloaded (QJSTCIOF).

This number reflects the number of active log CIs archived to the archive log medium.

**Tuning Tip:** This number will vary per time interval based upon the size of the active log data set. It is generally recommended that an active log be approximately the capacity of a cartridge data set (approximately 300 cylinders of 3390 DASD).

ARCHIVE LOG RD ALLOCATIONS

Number of archive log read allocations (QJSTALR).

This counts every allocation of an archive log for backout or recover operations.

**Tuning Tip:** High numbers indicate that there are not enough active logs properly sized to prevent archive log allocation. In a high performance environment, consider having active logs to hold 24 hours worth of work and archive the logs to media which can be shared in the event of an application failure involving recovery of multiple table spaces. The ideal value is zero.

ARCHIVE LOG WR ALLOCATIONS

Number of archive log write allocations (QJSTALW).

This counts every allocation of an archive log for write.

This is an indication of how many archive logs are written. Larger archive logs mean fewer allocations.

LOOK-AHEAD TAPE MOUNT ATT.

Number of look-ahead tape mounts attempted (QJSTLAMA).

DB2 attempts to pre-stage the second and third archive tape or cartridge volumes with archive data sets to reduce operator mount time. This counter reflects the number of times DB2 had to do this.

**Tuning Tip:** This number should be equal to the number of successful look-ahead tape mounts. Enough tape/cartridge drives must be available to DB2 on the MVS image to make use of this feature.

LOOK-AHEAD TAPE MOUNT SUCC

Number of successful look-ahead tape mounts (QJSTLAMS).

This counter is incremented when DB2 is able to pre-stage the second and third volumes of multiple tapes or cartridges in a backout or recovery operation.

**Tuning Tip:** This number should be equal to the number of look ahead tape mounts attempted. If they are not equal, a shortage of tape drives allocated to DB2 probably exists. To determine the amount of drives available, issue the -DIS ARCHIVE command; if too few drives are allocated in the *count* field, consider issuing the -SET ARCHIVE command to allocate a higher number, assuming the tape or cartridge resources are available.

LOG SUSPENDS FOR LOG WRITES

Number of times that a log manager request results in a suspend for a log record that is being written out to the log data sets (QJSTLSUS).

LOG WRITE I/O REQUESTS

Total number of log write I/O requests (Media Manager calls) (QJSTLOGW).

This value includes waits for copy1 and copy2 active log data set writes.

LOG CI WRITTEN

Total number of log CIs written (QJSTCIWR).

This value includes CI rewrites and copy1 and copy2 active log data set writes.

If a given CI is rewritten five times, this counter is incremented by five.

**Tuning Tip:** This counter, multiplied by 4KB and divided by the statistics interval in seconds, represents the number of bytes per second of log data written to the active log data sets. When this value exceeds 1MB/sec per log copy, attention should be paid to log data set I/O tuning.

LOG SERIAL WRITES

Number of serial log write I/O requests (QJSTSERW).

A serial log write I/O request occurs when DB2 rewrites a log CI that was previously written as a partial CI, in a dual logging environment.

This value includes copy1 and copy2 active log data set writes.

**Tuning Tip:** The difference between QJSTSERW and QJSTLOGW represents the number of parallel log write I/O requests. Typically, the first CI in a list of CIs to be written in one start I/O is written serially, and the remaining CIs are written in parallel to both copy1 and copy2 active log data sets. This value is meaningful only when DB2 runs in dual active log mode.

LOG ASYNC WRITES-THRESHOLD

Number of times that an asynchronous log write request was scheduled because the log write threshold was reached (QJSTTHRW).

**Tuning Tip:** It is recommended to use the default write threshold of 20 buffers.

LOG BUFFER PAGEDINS

Number of times that a log output buffer was paged in before it could be initialized (QJSTBPAG).

When a log output buffer is paged in before it is initialized, the log write latch is held.

**Tuning Tip:** A nonzero value could indicate that the log output buffer size is too large or there is insufficient real storage to back the log output buffer size.

## Statistics Long Report—Page 3

The fields in this section are organized in the following categories. For the highlights, see page 254.

### CPU TIMES (SECONDS)

This section shows CPU times for the five DB2 address spaces:

- MSTR
- DBM
- IRLM
- DIST

TOTAL
Total TCB and SRB CPU time for the address space being tracked.

TCB
TCB CPU time (QWSAEJST).

This field provides the CPU time for the address space being tracked.

The information in the CPU time generally represents time which is not allocated to threads and is therefore usually asynchronous overhead.

SRB
SRB CPU time (QWSASRBT).

This field accumulates the CPU time for all SRB tasks assigned to the address space being tracked.

Many DB2 events are scheduled as SRBs (service request blocks) for which time can be obtained. They are, therefore, DB2 address space overhead.

## QUERY PARALLELISM

MAX DEGREE OF PARALLELISM

Maximum degree of parallel processing executed (QXMAXDEG).

This counter is set to the high-water mark among all parallel groups executed with I/O or CP parallelism.

Degree measures the number of parallel processes (I/O streams in DB2 3.1 / 4.1, CP tasks in DB2 4.1) active for a parallel group.

PARALLEL GROUPS EXECUTED

Number of parallel groups executed (QXTOTGRP).

This counter reflects the total number of parallel groups executed.

**Tuning Tip:** This number can provide the tuner with a good idea when peak periods exist which might require more buffer pool size or a higher parallel processing percentage of a buffer pool might be necessary.

PLANNED DEGREE

Number of parallel groups executed at planned degree (QXNORGRP).

This counter is incremented when the number of parallel tasks (degree) at execution time equal the number of parallel tasks planned at BIND time.

**Tuning Tip:** The higher this number is, the better DB2 is tuned. This is the ideal situation.

REDUCED DEGREE - NO BUFFER

Number of parallel groups degraded due to buffer shortage (QXREDGRP).

This counter is incremented when the buffer pool does not have enough buffer storage to support as many degrees of parallel processing as had been planned. DB2 checks buffer allocations at both BIND and execution time. It assumes there will be buffers set aside for parallel processing. If at execution time, a similar number of buffers do not exist, DB2 degrades the parallel processes to a lesser degree or no parallelism.

**Tuning Tip:** Three parameters can be altered (ALTER BUFFERPOOL command) to resolve this situation. The overall size of the buffer pool is controlled by the VPSIZE (virtual pool size) parameter. The amount of sequential buffers is set by the VPSEQT (virtual pool sequential threshold) parameter. In the amount of buffers reserved for sequential processing, a reserve of buffers available for parallel processing must be maintained by the VPPSEQT (virtual pool parallel sequential threshold) parameter. If a significant number of parallel processes are degraded due to buffer shortage or contention, consider using a different buffer pool or altering the buffer pool used so that enough parallel sequential buffers are present.

FALLBACK - NO BUFFER

Parallel group fallback to sequential due to buffer shortage (QXDEGBUF).

This counter is incremented when the buffer pool does not have enough buffers to support parallel processing. DB2 checks buffer allocations at both BIND and execution time. It assumes there will be buffers set aside for parallel processing. If at execution time, a similar number of buffers do not exist, DB2 degrades the parallel processes to a lesser degree or no parallelism.

**Tuning Tip:** Three parameters can be altered (ALTER BUFFERPOOL command) to resolve this situation. The overall size of the buffer pool is controlled by the VPSIZE (virtual pool size) parameter. The amount of sequential buffers is set by the VPSEQT (virtual pool sequential threshold) parameter. In the amount of buffers reserved for sequential processing, a reserve of buffers available for parallel processing must be maintained by the VPPSEQT (virtual pool parallel sequential threshold) parameter. If a significant number of parallel processes are degraded due to buffer shortage or contention, consider using a different buffer pool or altering the buffer pool used so that enough parallel sequential buffers are present.

FALLBACK - NO ESA

Parallel group fallback to sequential due to lack of sort assist (QXDEGESA).

This counter is incremented when DB2 detects that the hardware sort assist facility is not present to logically partition the DB2 temporary DSNDB07 work files. The parallel sort operation falls back to sequential.

**Tuning Tip:** This situation occurs until the necessary hardware assist is purchased.

FALLBACK - CURSOR

Parallel group fallback to sequential due to updateable cursor (QXDEGCUR).

This counter is incremented when DB2 detects a cursor that is not clearly read-only and falls back from the planned parallel processing to sequential access.

**Tuning Tip:** To resolve this problem, the application program should have a cursor which is unambiguously read-only, with updates through another cursor or statement.

FALLBACK - ENCLAVE

Parallel group fallback to sequential due to enclave services unavailable (QXDEGENC).

This counter is incremented when DB2 detects that MVS 5.2 enclave support is unavailable to support parallel CP processing. The parallel group falls back to sequential.

**Tuning Tip:** MVS enclave support sets objectives for parallel tasks to perform within service goals set by management in the MVS Workload Manager. The DB2 parallel tasks run as enclave SRBs. The solution is to migrate this system to MVS 5.2 as soon as practical to do so.

PRLL GROUPS INTENDED

Number of parallel groups that DB2 intended to run across the data sharing group (QXXCBPNX).

This count is incremented on the parallelism coordinator only at run time (DB2 5.1 and later).

SINGLE DB2-COORDINATOR=NO

Number of parallel groups executed on a single DB2 due to one of the following reasons (QXCOORNO):

- When the plan or package was bound, the COORDINATOR subsystem parameter was set to YES, but the parameter is set to NO when the program runs.
- The plan or package was bound on a DB2 with the COORDINATOR subsystem parameter set to YES, but the program is being run on a different DB2 that has the COORDINATOR value set to NO.

(DB2 5.1 and later)

SINGLE DB2-ISOLATION=RR

Number of parallel groups executed on a single DB2 because the plan or package was bound with an isolation value of repeatable read or read stability (QXISORR) (DB2 5.1 and later).

DB2 MEMBER SKIPPED-NO BUF

Number of times the parallelism coordinator had to bypass a DB2 when distributing tasks because there was not enough buffer pool storage on one or more DB2 members (QXXCSKIP).

This field is incremented only on the parallelism coordinator, and it is incremented only once per parallel group, even though it is possible that more than one DB2 has a buffer pool shortage for that parallel group (DB2 5.1 and later).

**Tuning Tip:** The purpose of this count is to indicate that there are not enough buffers on one or more members. Therefore, this count is incremented only when the buffer pool is defined to allow parallelism. For example, if VPXPSEQT=0 on an assistant, DB2 does not send parallel work there, but this count is not incremented.

REFORMULATED GROUPS-CONFIG

Total number of parallel groups for which DB2 reformulated the parallel portion of the access path because the sysplex configuration at run time was different from the sysplex configuration at bind time (QXREPOP1).

This counter is incremented only by the parallelism coordinator at run time.

(DB2 6.1 and later)

REFORMULATED GROUPS-NO BUFFER

Total number of parallel groups for which DB2 reformulated the parallel portion of the access path because there was not enough buffer pool resource (QXREPOP2).

This counter is incremented only by the parallelism coordinator at run time.

(DB2 6.1 and later)

## ROUTINES

These fields apply only to DB2 4.1 and later.

SPROC-CALL STATEMENTS
Number of CALL statements executed (QXCALL).

This counter is incremented every time an application program issues an SQL CALL statement to execute a DB2 stored procedure.

SPROC-ABENDS
Number of stored procedure abends (QXCALLAB).

This counter is incremented when a stored procedure abends.

**Tuning Tip:** Large numbers here indicate that one or more stored procedures have fatal errors.

SPROC-TIMEOUTS
Number of CALL statements timed out (QXCALLTO).

This counter is incremented when the Stored Procedures Address Space does not have a TCB available within the time limit specified in the DSNZPARM TIMEOUT parameter to schedule the execution of an SQL CALL statement.

Appropriate fixes include

- Reduce the amount of CPU service units in other stored procedures to get better throughput.
- Stop stored procedures no longer needed.
- Increase the number of TCBs available in the Stored Procedures Address Space (by increasing the NUMTCB parameter in the Stored Procedures Address Space JCL).

SPROC-REJECTS
Number of CALL statements rejected because procedure was stopped (QXCALLRJ).

This counter is incremented when an SQL application CALLs a stored procedure which has been stopped by the operator or system administrator.

**Tuning Tip:** If this happens frequently, be sure the procedure is started in normal mode.

TRIGGER-STMT ACTIVATED
Number of times a statement trigger is activated (QXSTTRG).

(DB2 6.1 and later)

TRIGGER-ROW ACTIVATED
Number of times a row trigger is activated (QXROWTRG).

(DB2 6.1 and later)

TRIGGER-SQL ERRORS
Number of times an SQL error occurred during execution of a triggered action (QXTRGERR).

(DB2 6.1 and later)

UDF -EXECUTIONS
Number of user-defined functions executed (QXCAUD).

(DB2 6.1 and later)

UDF -ABENDS
Number of times a user-defined function abended (QXCAUDAB).

(DB2 6.1 and later)

UDF -TIMEOUTS
Number of times a user-defined function timed out waiting to be scheduled (QXCAUDTO).

(DB2 6.1 and later)

UDF -REJECTS
Number of times a user-defined function was rejected (QXCAUDRJ).

(DB2 6.1 and later)

## PLAN/PACKAGE PROCESSING

ALLOCATE PLAN ATTEMPTS

Plan allocation attempts (QTALLOCA).

This counter is incremented each time a calling program attempts to allocate a bound plan to create a thread.

Plan allocation occurs at the first SQL statement in the unit of work. Plan allocation results in the necessary sections of the skeleton cursor/skeleton package tables being loaded into the EDM pool if not already present.

ALLOCATE PLAN SUCCESSFUL

Number of successful bound plan allocations (QTALLOC).

This counter is incremented every time a bound plan is successfully allocated.

**Tuning Tip:** Allocation may fail if the plan is locked (such as during another program BIND) or if the plan name is nonexistent.

ALLOCATE PACKAGE ATTEMPTS

Package allocation attempts (QTPKALLA).

This counter is incremented each time a calling program attempts to allocate a bound package.

Package allocation occurs at the first SQL statement in the package. Package allocation results in the necessary sections of the skeleton package tables being loaded into the EDM pool if not already present.

ALLOCATE PACKAGE SUCCESSFUL

Number of successful package allocations (QTPKALL).

This counter is incremented every time a bound package is successfully allocated.

Packages allocation may fail if the relevant pages were locked (such as during another program BIND) or the package name is nonexistent.

AUTOBIND ATTEMPTS

Number of automatic BINDs attempted (QTABINDA).

This counter is incremented when DB2 attempts to automatically bind a plan that has been rendered invalid by an operation such as dropping an index.

**Tuning Tip:** If the DSNZPARM parameter ABIND is set to YES (the default), DB2 attempts to automatically rebind plans which have been rendered invalid by the dropping of a resource necessary to fulfill an access path. This BIND operates under SYSOPR command authority. If successful, the SQL may use different access paths; however, automatic rebind can fail due to lock timeouts if other pages in the catalog currently have locks pending. A plan cannot be run while being bound.

AUTOBINDS SUCCESSFUL

Number of successful automatic BINDs (QTABIND).

This counter is incremented if an automatic rebind is successful.

Automatic rebinds may be unsuccessful if necessary tables have been dropped or if resources are unavailable, such as locks existing on the directory and/or catalog.

AUTOBIND INV. RESOURCE IDS
Automatic BIND with invalid resource IDs (QTINVRID).

This counter is incremented every time an automatic rebind occurs when a necessary structure (such as a table or table space) is no longer present.

This is the number of failed plan or package allocation attempts caused by missing or unavailable resources. When automatic rebind fails, a message is normally produced in the MSTR job log with the reason code for the failure.

AUTOBIND PACKAGE ATTEMPTS
Number of automatic BIND PACKAGEs attempted (QTAUTOBA).

This counter is incremented when DB2 attempts to automatically bind a package that has been rendered invalid by an operation such as dropping an index.

**Tuning Tips:** If the DSNZPARM parameter ABIND is set to YES (the default), DB2 attempts to automatically rebind packages or plans that have been rendered invalid by the dropping of a resource necessary to fulfill an access path. This bind operates under SYSOPR command authority. If successful, the SQL may use different access paths. However, automatic rebind can fail due to lock timeouts if other pages in the catalog currently have locks pending. A plan/package be cannot run during the bind process.

AUTOBIND PACKAGE SUCCESSFUL
Number of successful automatic BIND PACKAGEs (QTPKABND).

This counter is incremented if an automatic rebind of a package is successful.

**Tuning Tip:** Automatic rebinds may be unsuccessful if necessary tables have been dropped or if resources are unavailable, such as locks existing on the directory and/or catalog.

PLANS BOUND
Number of plans successfully bound (QTPLNBD).

This counter is incremented each time plans are bound and stored for later allocation.

This reflects the total number of BIND ADD and REPLACE operations that have been done.

BIND ADD SUBCOMMANDS
Number of BIND ADD subcommands issued (QTBINDA).

This counter is incremented when a BIND ADD is attempted, whether successful or not.

**Tuning Tip:** Bind activity should be monitored so it does not take place during prime production hours. This information can help determine when BIND activity occurs.

BIND REPLACE SUBCOMMANDS
Number of BIND REPLACE subcommands issued (QTBINDR).

This counter is incremented when a BIND REPLACE is attempted, whether successful or not.

**Tuning Tip:** Bind activity should be monitored so it does not take place during prime production hours. This information can help determine when BIND activity occurs.

TEST BIND SUBCOMMANDS

Number of BIND subcommands issued without a plan ID (QTTESTB).

This count is incremented each time a BIND (*) is successfully completed.

**Tuning Tip:** Normally, BIND should be run with a valid plan name.

PACKAGES BOUND

Number of packages successfully bound (QTPKGBD).

This number is incremented for each package that is successfully bound and stored for later allocations.

The packages are stored in the SPT01 directory table.

BIND ADD PACKAGE

Number of BIND ADD PACKAGE subcommands issued (QTBINDPA).

This counter is incremented for each BIND ADD PACKAGE commands issued.

This number includes successful and unsuccessful BIND ADD PACKAGE counts.

BIND REPLACE PACKAGE

Number of BIND REPLACE PACKAGE subcommands issued (QTBINDPR).

This counter is incremented for each BIND REPLACE PACKAGE command issued.

This number includes successful and unsuccessful BIND REPLACE PACKAGE counts.

REBIND SUBCOMMANDS

Number of REBIND subcommands issued (QTREBIND).

This number is incremented for every REBIND issued.

REBIND rebuilds the access path without a change in program or SQL code; BIND REPLACE implies a change in code.

REBIND PLAN ATTEMPTS

Number of attempts to rebind a plan (QTRBINDA).

This number is incremented with the number of attempts to rebind a plan.

This number can be more than one plan per rebind command since wildcarding is allowed, and that could rebind many plans.

PLANS REBOUND

Number of times a plan was successfully rebound (QTPLNRBD).

This counter is incremented each time a REBIND of a plan is successful.

**Tuning Tip:** Typical reasons for not being successful are catalog/directory locks (the plan is running when the attempt to REBIND is made) or a necessary structure (table or table space) is not available.

REBIND PACKAGE SUBCOMMANDS

Number of REBIND PACKAGE subcommands issued (QTRBINDP).

This counter is incremented for each REBIND PACKAGE command issued.

More than one package can be rebound in a single command.

REBIND PACKAGE ATTEMPTS

Number of attempts to REBIND a package (QTRBNDPA).

This counter is incremented once for each attempt to REBIND a package.

More than one package can be rebound in a single command.

PACKAGES REBOUND

Number of packages successfully rebound (QTPKGRBD).

This number is incremented for each package that is successfully rebound.

The packages are stored in the SPT01 directory table.

FREE PLAN SUBCOMMANDS

Number of FREE subcommands issued (QTFREE).

This number is incremented every time a FREE command to delete one or more plans is issued.

Plans cannot be FREEd while executing.

FREE PLAN ATTEMPTS

Number of attempts to FREE a plan (QTFREEA).

This counter is incremented for each plan attempted to be freed.

More than one plan can be FREEd in a single command.

PLANS FREED

Number of plans FREEd (QTPLNFRD).

This number represents the total number of FREE plan attempts that complete successfully.

Plans cannot be FREEd while executing.

FREE PACKAGE SUBCOMMANDS

Number of FREE PACKAGE subcommands issued (QTFREEP).

This counter is incremented once for each FREE PACKAGE command issued.

More than one package can be FREEd in a single command.

FREE PACKAGE ATTEMPTS

Number of attempts to FREE a package (QTFREEAP).

This counter is incremented once for each attempt to FREE a package.

More than one package can be FREEd in a single command.

PACKAGES FREED

Number of packages FREEd (QTPKGFRD).

This number represents the total number of FREE package attempts that complete successfully.

Packages cannot be FREEd while executing.

INCREMENTAL BINDS

Number of incremental BINDs performed (QXINCRB).

This counter is incremented every time a plan is run that had been bound with the VALIDATE(RUN) option.

**Tuning Tip:** It is generally undesirable to bind a plan with the VALIDATE(RUN) option since all SQL statements must be rechecked for syntax, authority, and access path every time the plan is executed. VALIDATE(RUN) is required if the program is going to CREATE TABLES (for example, work tables) during the execution or if testing is required on a piece of code for which the objects do not yet exist. Otherwise, VALIDATE(RUN) should be avoided, as the cost is nearly that of dynamic SQL.

# Statistics Long Report—Page 4

The fields in this section are organized in the following categories. For the highlights, see page 254.

## GLOBAL BUFFERPOOL

This section is available for DB2 4.1 and later only.

SYNC RD INV BUFF
Number of synchronous coupling facility reads for *invalid buffer* in each of these circumstances:

- WITH DATA
A data sharing coupling facility synchronous read was issued because a local virtual buffer pool or hiperpool had a page marked as invalid. The page existed in the group buffer pool and was returned (QBGLXD).

This occurs normally in a data sharing complex in which data must be shared. The correct data is refreshed from the group buffer pool.

- NO DATA
A data sharing coupling facility synchronous read was issued because a local virtual buffer pool or hiperpool had a page marked as invalid but no data was returned (QBGLXR + QBGLXN (DB2 Version 4 only)).

SYNC RD NOT FOUND
Number of synchronous coupling facility reads for *buffer not found* in each of these circumstances:

- WITH DATA
A data sharing coupling facility synchronous read was issued because a page was not found in the local virtual buffer pool or hiperpool but the page existed in the group buffer pool and was returned (QBGLMD).

This occurs normally in a data sharing complex in which data must be shared. The correct data is refreshed from the group buffer pool.

- NO DATA
A data sharing coupling facility synchronous read was issued because a page could not be found in a local virtual buffer pool or hiperpool and no data was returned (QBGLMR + QBGLMN (DB2 Version 4 only)).

ASYNC READ
Number of asynchronous coupling facility reads in each of these circumstances:

- DATA RETURNED
A page existed in a global buffer pool and was returned (QBGLAD).

This is on behalf of prefetch. It occurs normally in a data sharing complex in which data must be shared. The correct data is refreshed from the group buffer pool.

- NO DATA
A data sharing coupling facility asynchronous read was issued but no data was returned from the group buffer pool (QBGLAR + QBGLAN (DB2 Version 4 only)).

This is on behalf of prefetch.

SYNC PAGES WRITTEN

Number of synchronous pages written to the group buffer pool in each of these circumstances:

-CHANGED

A changed page was written to the group buffer pool (QBGLSW).

Updated pages must be written to the coupling facility when the object is of intersystem interest so other DB2s can refresh their invalidated buffers. Only changed pages are written when GBPCACHE is set to the default of CHANGE.

-CLEAN

A clean page was written to the group buffer pool (QBGLWC).

This is done only when GBPCACHE is set to ALL.

**Tuning Tip:** This can be expensive but it does post the group buffer pool with clean pages and may be useful for small tables with high intersystem read interest to reduce contention.

ASYNC PAGES WRITTEN

Number of asynchronous pages written to the group buffer pool in each of these circumstances:

-CHANGED

A local changed page was written asynchronously to the group buffer pool (QBGLAW).

This is initiated by normal update processing in a data sharing complex.

-CLEAN

A clean page was written asynchronously to the group buffer pool (QBGLAC).

This is done for page sets and partitions defined with GBPCACHE ALL. All clean pages can be kept in the coupling facility to improve access to data sets with high contention.

CASTOUT

-PAGES WRITTEN

Number of pages cast out from the GBP (QBGLRC).

A data sharing complex periodically externalizes updated pages to DASD in a process known as castout.

The castout data is externalized for various reasons:

- A GBP castout threshold is reached (similar to a virtual buffer pool reaching its deferred write threshold)
- A class castout threshold is reached (similar to a data set reaching its vertical deferred write threshold)
- A GBP checkpoint is triggered
- There is no more inter-DB2 interest in the page set

-NO ENGINE

Number of times a castout engine was unavailable (QBGLCN).

This counter is incremented when the coupling facility could not successfully cast out its pages because enough castout engines were not available.

Workloads need to be shifted to make sure the coupling facility can successfully externalize its data. DB2 casts out the pages when an engine becomes available.

-CLASS THRESHOLD

Number of times GBP castout was initiated because the class castout threshold was reached (QBGLCT).

Data Sharing GBP castout processing is initiated because the number of changed pages for a castout class queue exceeded the class threshold. This causes updated pages to be read by the *owning* DB2 from the GBP and written to DASD.

**Tuning Tip:** This process is similar to vertical deferred write processing on a local DB2 and is a normal way to externalize updated pages. The default class castout threshold is a single class (one or more page sets / partitions) occupying 10% or more of the group buffer pool. This threshold is tunable.

-GBP THRESHOLD

Number of times GBP castout was initiated because the GBP castout threshold was reached (QBGLGT).

Data Sharing GBP castout processing is initiated because the number of changed pages in the GBP exceeded the GBP castout threshold. This causes updated pages to be read by the *owning* DB2 from the GBP and written to DASD.

**Tuning Tip:** This process is similar to deferred write processing on a local DB2 and is a normal way to externalize updated pages. The default GBP castout threshold is 50%. This threshold is tunable.

NO WRITE ENGINE

Number of times a coupling facility write engine was unavailable (QBGLSU).

This counter is incremented when the coupling facility exceeds the number of write engines required.

**Tuning Tip:** This number should be extremely low. It may be necessary to reduce the number of page sets being shared.

READ FAIL -STORAGE

Number of failed coupling facility reads due to lack of storage (QBGLRF).

This counter is incremented when there is not enough storage allocated to the coupling facility to manage that required by the group buffer pools.

**Tuning Tip:** This indicates that the group buffer pools are overcommitted. Reduce the size of group buffer pools unless more coupling facility memory can be acquired to support the group buffer pools.

WRITE FAIL -STORAGE

Number of failed coupling facility writes due to lack of storage (QBGLWF).

This counter is incremented when there is not enough storage allocated to the coupling facility to complete all write requests.

**Tuning Tip:** The group buffer pools are overcommitted. Reduce the size of the group buffer pools unless more coupling facility memory can be acquired to support them.

RD STG STATS(5.1)/OTHER(4.1)

Number of other coupling facility interactions experienced during castout processing (QBGLOS).

GBP CHECKPOINT

Number of group buffer pool checkpoints triggered by this member (QBGLCK) (DB2 5.1 and later).

GBP REBUILD

Number of group buffer pool rebuilds in which this member participated (QBGLRB) (DB2 5.1 and later).

UNLOCK CASTOUT

Number of coupling facility requests to unlock the castout lock for castout I/Os that have completed (QBGLUN).

As pages are in the process of being cast out to DASD, they are *locked for castout* in the coupling facility. The castout lock is not an IRLM lock; its purpose is to enforce that only one system is doing castout for a given page at a time (DB2 5.1 and later).

**Tuning Tip:** DB2 usually includes multiple pages in a write I/O request for castout. Therefore, this value should always be less than or equal to the number of pages cast out. It will be significantly less if multiple pages are written per I/O, so this ratio gives a good indication of how castout is performing.

READ CASTOUT CLASS

Number of coupling facility requests to read the castout class (QBGLCC).

DB2 uses this number to determine which pages that belong to a given page set or partition are cached as *changed* in the group buffer pool and need to be cast out (DB2 5.1 and later).

This request is issued by the page set of partition castout owner. It is also issued by the group buffer pool structure owner when the GBPOOLT threshold is reached.

READ CASTOUT STATISTICS

Number of coupling facility requests to read the castout statistics (QBGLCS).

The group buffer pool structure owner issues this request when the group buffer pool threshold GBPOOLT is reached to determine which castout classes have changed pages. Generally, this occurs only once or twice for each occurrence of the GBPOOLT threshold being reached (DB2 5.1 and later).

DELETE DIR/DATA ENTRIES

Number of group buffer pool requests to delete all directory and data entries for a page set or partition (QBGLDN).

DB2 issues this request when it converts a page set or partition from GBP-dependent to non-GBP-dependent. For objects defined with GBPCACHE ALL, DB2 issues this request when the first DB2 member opens the object (DB2 5.1 and later).

READ DIRECTORY INFORMATION

Number of coupling facility requests to read directory information (QBGLRD) (DB2 5.1 and later).

The group buffer pool structure owner issues this request for GBP checkpoints to read the directory entries of all changed pages in the group buffer pool so that the oldest recovery LRSN can be recorded and used for recovery purposes in case the group buffer pool should fail. (This recovery LRSN is displayed in message DSNB798I.)

**Tuning Tip:** This request may need to be issued several times for each GBP checkpoint in order to read the directory entries for all changed pages. If you see an abnormally high number here, it might be that these requests are being cut short by the coupling facility model-dependent timeout criteria. To help alleviate this, you should consider upgrading those coupling facilities to CFLEVEL 2 or above.

REGISTER PAGE

Number of coupling facility requests to register interest to the GBP for a single page (QBGLRG).

These are *register-only* requests. DB2 is not requesting that any data be returned for the page because it knows that the page is not cached in the group buffer pool (DB2 5.1 and later).

The *register page* request is made only to create a directory entry for the page for cross-invalidation when downgrading the P-lock on a page set or partition from S mode to IS mode, or from SIX mode to IX mode.

UNREGISTER PAGE

Number of coupling facility requests to unregister interest to the GBP for a single page (QBGLDG).

This is generally done as DB2 steals pages from the local buffer pool that belong to GBP-dependent page sets or partitions (DB2 5.1 and later).

REGISTER PAGE LIST

Number of requests to register a page list in the coupling facility (QBGLAX).

DB2 prefetch can do this only if the member is running on MVS 5.2 or later and if the group buffer pool is allocated in a coupling facility with CFLEVEL=2 or above (DB2 4.1 and later).

REGISTER PAGE LIST-RD CHNGE

Number of coupling facility reads to retrieve a changed page for a page list (QBGLAY).

This is the number of coupling facility reads to retrieve a changed page from the group buffer pool as a result of feedback from the request to register a page list (DB2 4.1 and later).

DB2 prefetch can do this only if the member is running on MVS 5.2 or later and if the group buffer pool is allocated in a coupling facility with CFLEVEL=2 or above.

REGISTER PAGE LIST-RD CLEAN

Number of coupling facility requests to retrieve a clean page for a page list (QBGLAZ).

This is the number of coupling facility reads to retrieve a clean page from the group buffer pool as a result of feedback from the request to register a page list (DB2 4.1 and later).

DB2 prefetch can do this only if the member is running on MVS 5.2 or later and if the group buffer pool is allocated in a coupling facility with CFLEVEL=2 or above.

EXPLICIT CROSS INVALIDATES

Number of explicit cross-invalidations (QBGLEX).

(DB2 6.1 and later)

DUPLEX-WRITE REQUEST

Number of coupling facility requests to write changed pages to the secondary group buffer pool for duplexing (QBGL2W).

(DB2 6.1 and later)

DUPLEX-WRITE FAIL

Number of coupling facility requests to write changed pages to the secondary group buffer pool for duplexing that failed due to a lack of storage in the coupling facility (QBGL2F).

(DB2 6.1 and later)

DUPLEX-DELETE NAME LIST

Number of group buffer pool requests to the secondary group buffer pool to delete a list of pages after they have been cast out from the primary group buffer pool (QBGL2D).

(DB2 6.1 and later)

DUPLEX-DELETE NAME

Number of group buffer pool requests to delete a page from the secondary group buffer pool (QBGL2N).

These requests are issued by the group buffer pool structure owner to delete orphaned data entries in the secondary as part of the garbage collection logic.

(DB2 6.1 and later)

DUPLEX-READ CASTOUT STATS

Number of coupling facility requests to read the castout statistics for the secondary group buffer pool (QBGL2R).

These requests are issued by the group buffer pool structure owner to check for orphaned data entries in the secondary.

(DB2 6.1 and later)

## GLOBAL LOCKING ACTIVITY

This section is available for DB2 4.1 and later only.

PLOCK REQUESTS

-LOCK

Number of lock requests for P-locks (QTGSLPLK).

This counter is incremented when a physical lock (P-lock) is acquired in the DB2 data sharing environment at the data set level to allow DB2 to determine whether an object is of inter-DB2 interest. Concurrency is still managed as before with transaction (or logical) locks (L-locks).

A P-lock is acquired for the first transaction that accesses an object (table space, index space, partition) but it is owned by the subsystem. Page P-locks are used for row-level locking. Inter-DB2 interest occurs only when at least one DB2 has update interest and one or more others have read interest. P-locks never cause timeouts, as the lock mode can be negotiated.

-CHANGE

Number of change requests for P-locks (QTGSCPLK).

This counter is incremented when a P-lock has been established and the IRLM associated with this DB2 determines that the nature of that lock must be changed (for example from share to exclusive) because the usage of the page set in that DB2 has changed.

The number of P-lock changes reflects the changing status of page sets as access modes change and the lock state is negotiated between data sharing members.

-UNLOCK

Number of unlock requests for P-locks (QTGSUPLK).

This number represents the number of IRLM unlock requests because a page set is no longer in use.

This number represents the number of times a page set has been unlocked in a data sharing environment.

XES SYNC REQ

-LOCK

Number of XES synchronous lock requests (QTGSLSLM).

This number is incremented for both L-locks and P-locks when a request for a lock on an object of inter-DB2 interest is sent to MVS Cross-System Services. This occurs synchronously under the user's execution unit.

This number reflects the request for locks of inter-DB2 interest after it is determined that no intrasystem locks exist. This count is not incremented if a suspension occurs.

-CHANGE

Number of XES synchronous change requests (QTGSCSLM).

This number is incremented for both L-locks and P-locks when a change request for a lock on an object of DB2 intersystem interest is sent to MVS Cross-System Services.

This number reflects the number of change requests for locks of inter-DB2 interest. This count is not incremented if a suspension occurs.

-UNLOCK

Number of XES synchronous unlock requests (QTGSUSLM).

This number is incremented for both L-locks and P-locks when an unlock request to an object of inter-DB2 interest is sent to MVS Cross-System Services. It is incremented for each resource for which the global lock is released, rather than once for a generic unlock request as is done for local locks.

This number reflects all locks which are released through MVS Cross-System Services. This count is not incremented if a suspension occurs.

SUSPENDS

-IRLM

Number of global suspends - IRLM (QTGSIGLO).

This number is incremented when IRLM global lock states were in conflict on the same resource (page set, page, row), where the lock was held by another DB2 data sharing group member.

**Tuning Tip:** This counter is a measure of inter-DB2 contention. All locks of intrasystem contention are resolved prior to the IRLM checking global contention. This is the number of true waits for another thread on a different MVS that has an incompatible lock type to the requester. Global contention requires intersystem communication to resolve the conflict. Data sharing is designed to optimize the quick granting of locks where no contention exists. Try to keep total contentions (IRLM, XES, and false) to less than 2% of the total lock requests.

-XES

Number of global suspends - XES (QTGSSGLO).

This counter is a measure of MVS XES global resource contention.

MVS XES lock states were in conflict but IRLM lock states were not. XES is aware only of exclusive or share locks, where IRLM has many lock states.

-FALSE

Number of global suspends due to false contention (QTGSFLSE).

This counter is incremented every time XES detects contention on the resource hash class but there was no real contention on the resource itself.

**Tuning Tip:** MVS Cross-System Services uses a hash lock table in the coupling facility to provide efficient determination of whether a resource is locked. This hash table points to a number of synonyms (different resource names can hash to the same entry). Contention at the hash entry is considered false contention. The more resources declared to be of inter-DB2 interest, the more chances there are of this condition occurring. If false contentions are more than half of the total global contentions (suspensions for IRLM, XES and false), the CF lock structure size may be too small for the workload.

GLOBAL LOCK CONTENTION(%)

Percentage of global lock contentions.

This value is calculated as

Global Suspends / Total Global Locks * 100.0

where

Global Suspends is the total number of global lock contentions (QTGSIGLO + QTGSSGLO + QTGSFLSE).

Total Global Locks is the sum of

- Lock requests (QTGSLSLM)
- Change requests (QTGSCSLM)
- Unlock requests (QTGSUSLM)
- Global lock contentions (QTGSIGLO + QTGSSGLO + QTGSFLSE)

FALSE CONTENTION(%)

Percentage of global lock false contentions.

This value is calculated as

False Contentions / Global Suspends x 100.0

where

False Contentions is the total number of global lock suspensions due to false contention (QTGSFLSE).

Global Suspends is the total number of global lock contentions (QTGSIGLO + QTGSSGLO + QTGSFLSE).

INCOMPATIBLE RETAINED LOCK

Number of global requests denied due to incompatible retained lock (QTGSDRTA).

This number is incremented every time global lock services denies a lock request because an incompatible lock type has been retained on the requested resource.

This number reflects the instances in which access to a page or row cannot be obtained because another thread on another system had access to the resource requested but the other system failed. The CF retains locks until the other system comes up to complete the unit of work.

NOTIFY MESSAGES -SENT

Number of notify messages sent (QTGSNTFY).

This number represents the number of messages passed to the coupling facility so that, for example, other members can be notified that DBDs have been changed due to CREATE, ALTER, or DROP.

This is the outbound traffic to the coupling facility.

NOTIFY MESSAGES -RECEIVED

Number of notify messages received (QTGSNTFR).

This number represents the number of messages received from the coupling facility.

This is the inbound traffic from the coupling facility.

XES ASYNC REQ -RESOURCES

Number of resources propagated asynchronously to XES (QTGSKIDS).

This counter is incremented whenever a DB2 data sharing P-lock or L-lock (page set lock or row/page lock) is propagated by the IRLM to MVS Cross-System Services asynchronously.

This situation occurs when new inter-DB2 interest occurs on a parent resource (such as a page set) or when a request completes after the requester's execution unit has already been suspended.

NEGOTIATE

-PGSET/PARTITION

Number of P/P P-lock negotiations (QTGSPPPE).

This counter is incremented when the participating DB2 has to change the lock type of a page set / partition P-lock because another thread on another system changed its intent on this object from SHARE to EXCLUSIVE or from EXCLUSIVE to SHARE.

Each DB2 in a data sharing group may need to change P-locks as part of a negotiation process so that physical locks never cause timeouts or deadlocks.

-PAGE PLOCK

Number of Page P-lock negotiations (QTGSPGPE).

This counter is incremented when the participating DB2 has to change a page lock type because another thread on another system has changed its intent on this object from SHARE to EXCLUSIVE or EXLUSIVE to SHARE. Page P-locks are used for row-level locking.

Each DB2 in a data sharing group may need to change P-locks as part of a negotiation process so that physical locks never cause timeouts or deadlocks.

-OTHER LOCK

Number of other P-lock negotiations (QTGSOTPE).

This counter is incremented when anything other than a page set or page lock is negotiated between participating DB2 subsystems.

These events represent other contention which may exist other than page set or page P-locks.

-PLOCK CHANGE REQ

Number of P-lock change requests during P-lock negotiation (QTGSCHNP).

This counter is incremented when a participating DB2 must communicate a lock change request to other DB2s as a result of negotiation for access to an object.

This is a count of the number of P-Lock changes issued during P-Lock intersystem negotiation.

PLOCK/NOTIFY

-MAX ENGINES

Maximum number of engines available for lock/notify exit requests (QTGSPEMX).

This number reflects the number of tasks allocated to tracking intersystem lock negotiation.

This is a count of the facilities available to deal with lock contention from the global DB2 data sharing perspective.

-NO ENGINE

Number of times no engine available for lock/notify exit requests (QTGSPEQW).

This number is incremented when the number of requests for lock or notify exit processing exceeds the maximum number of engines available.

**Tuning Tip:** This number indicates that there are too many intersystem locking and notification requirements based on the number of resources available for global lock management. It may be appropriate to reduce the number of resources being shared.

## GLOBAL DDF ACTIVITY

DBATS QUEUED(AT MAX)
Number of times a DBAT thread was queued (QDSTQDBT).

This counter indicates that a requester thread was queued because the serving system did not allow enough active remote threads. The tuning parameter is MAXDBAT in DSNZPARM.

**Tuning Tip:** If this situation occurs often, increase MAXDBAT in DSNZPARM on installation panel DSNTIPE, reassemble, and restart DB2.

CONVERSAT. DEALLOC(AT MAX)
Number of conversations deallocated due to ZPARM limit (QDSTQCRT).

This situation occurs when the thread count from remote locations exceeds the sum of maximum active threads and inactive threads in DSNZPARM.

**Tuning Tip:** If this occurs often, update the ‘MAX REMOTE CONNECTED’ (CONDBAT) field in DSNZPARM on installation panel DSNTIPE, reassemble, and restart DB2. The maximum in DB2 3.1 is 10,000; in DB2 4.1 it is 25,000.

INACTIVE DBATS-CURRENT
Current number of inactive DBATs (QDSTQCIT).

This counter reflects a snapshot count of inactive requester threads to this server.

An inactive thread is a thread that is currently not executing but the VTAM connection has been kept intact on the assumption this user will still be doing further work.

INACTIVE DBATS-MAX
Maximum number of inactive DBATs (QDSTQMIT).

This counter reflects the high-water mark of inactive threads.

**Tuning Tip:** If the number is close to the value in DSNZPARM, consider increasing the number of inactive threads allowed (CONDBAT).

ACTIVE DBATS-CURRENT
Current number of active DBATs (QDSTCNAT).

ACTIVE DBATS-MAX
Maximum number of active DBATs that existed (QDSTHWAT).

TOTAL DBATS-MAX
Maximum number of remote connections that existed (QDSTHWDT).

COLD START CONNECTIONS

Number of cold start connections with all remote locations (QDSTCSTR).

This reflects DB2 attempting to synchronize with a requester which has been cold-started (two-phase commit operations only).

**Tuning Tip:** The primary significance of this number is that if it is non-zero, check whether any indoubt threads were left unresolved by the cold start. If so, the system programmer must issue -RESOLVE INDOUBT (COMMIT or ABORT) commands.

WARM START CONNECTIONS

Number of warm start connections with all remote locations (QDSTWSTR).

This counter indicates all connections made through normal restart processing (two-phase commit operations only).

**Tuning Tip:** This number is of interest in determining that a problem does ***not*** exist.

RESYNC ATTEMPTS

Number of resynchronization attempts with all remote locations (QDSTRSAT).

This counter is incremented when communication is lost between a requester and server (two-phase commit operations only). Following restoration of communications, DB2 attempts to resynchronize all servers and requesters.

A large number of resynchronization attempts indicates network difficulties or MVS problems.

RESYNC SUCCESSES

Number of successful resynchronization attempts (QDSTRSSU).

This counter is incremented when DB2 has successfully completed the resolution of indoubt processing after a communications failure.

High numbers in this counter indicate VTAM problems that should be resolved with the VTAM system programmer.

## AUTHORIZATION MANAGEMENT

PLAN - AUTHORIZATION CHECKS
Number of plan authorization check attempts (QTAUCHK).

This counter is incremented every time a plan authorization check is performed.

The authorization check takes place by looking first in the EDM pool cache (specified as CACHESIZE on the BIND operation) and then doing catalog authorization lookups.

PLAN - SUCC CHECKS
Number of successful authorization checks (QTAUSUC).

This counter is incremented each time a plan is checked with a successful result.

When the first successful check is done, the AUTHID is stored in the EDM pool SKCT cache. The next attempt does the lookup in the cache before going to the catalog.

PLAN - SUCC CHECKS - PUBLIC
Number of successful authorization checks based on EXECUTE authority GRANTed to PUBLIC (QTAUPUB).

This counter is incremented each time the plan’s authorization was GRANTed to PUBLIC. This is the fastest form of access.

**Tuning Tip:** This number is useful in determining which mechanism is allowing authorization. GRANTing to PUBLIC is the fastest since a special bit is set that disables further checking.

PLAN - SUCC CHECKS - CACHE
Number of successful authorization checks not using the catalog (QTAUCCH).

This number is incremented every time a plan was GRANTed to PUBLIC or the authorization ID is found in the EDM pool cache.

**Tuning Tip:** If not granting plans to public, it is desirable to have a large cache size or to use secondary group authorizations.

PKG - SUCC CHECKS - CACHE
Number of times a successful check for package execute authority is made without accessing the catalog (QTPACAUT).

This number also includes cache hits, as well as public authorization checks (DB2 5.1 and later).

PKG - UNSUCC CHECKS - CACHE
Number of times an unsuccessful check for package execute authority is made because an applicable entry was not found in the package authorization cache, requiring catalog access (QTPACNOT).

(DB2 5.1 and later)

PKG - SUCC CHECKS - PUBLIC
Number of times a successful check for package execute authority is made for a package GRANTed to PUBLIC (QTPACPUB).

(DB2 5.1 and later)

OVERWROTE AUTHID - CACHE
Number of times an entry for an authorization ID was overwritten in the package authorization cache (QTPACOW1).

(DB2 5.1 and later)

OVERWROTE PKG/COLLID - CACHE
Number of times an entry for a package or collection was overwritten in the package authorization cache (QTPACOW2).

(DB2 5.1 and later)

ROUTINE CACHE-SUCCESS
Number of successful authorization checks for user-defined functions or stored procedures that use the routine authorization cache (QTRACAUT).

This number includes the number of public authorization checks (QTRACPUB).

(DB2 6.1 and later)

ROUTINE CACHE-SUCCESS PUBLI
Number of successful authorization checks for user-defined function or stored procedure execution authority when that authority is held by public (QTRACPUB).

(DB2 6.1 and later)

ROUTINE CACHE-NO CACHE
Number of authorization checks for user-defined function or stored procedure execution authority that could not use the routine authorization cache (QTRACNOT).

(DB2 6.1 and later)

ROUTINE CACHE-OVERWRITE AUT
Number of times that DB2 wrote over an authorization ID in the routine authorization cache (QTRACOW1).

(DB2 6.1 and later)

ROUTINE CACHE-OVERWRITE ENT
Number of times that DB2 wrote over a routine entry in the routine authorization cache (QTRACOW2).

(DB2 6.1 and later)

ROUTINE CACHE-ENTRY NOT ADD
Number of times that DB2 could not add an entry to the routine authorization cache (QTRACNAC).

(DB2 6.1 and later)

## Statistics Long Report—Page 5

The fields in this section are organized in the following categories. For the highlights, see page 254.

### BP GENERAL

CURRENT ACTIVE BUFFERS

Number of current active (non-stealable) buffers (QBSTCBA).

This is a snapshot look at the current number of pages that are in use or updated and not yet written to DASD.

**Tuning Tip:** If the number of non-stealable buffers is regularly greater than 50%, consider increasing the virtual pool size. If greater than 80%, the pool is in danger of reaching the 90% sequential prefetch shutoff mechanism. In a typical online system, this number tends to be small. But in a batch or combined online and batch environment, this number can become large if any application fails to commit. If increasing the size of the pool is not a viable option (because no memory exists to support it), consider increasing the commit frequency of applications to free committed pages.

VP BUFFER POOL FULL

Number of times virtual buffer pool was full (QBSTXFL).

This counter is incremented every time DB2 attempts to perform a GETPAGE but could not find a free buffer in the virtual buffer pool because it was full of updated or in-use pages.

**Tuning Tip:** This is a worst-case situation. DB2 will have hit many other thresholds which cause poor performance before this occurs. This number should always be zero. If a buffer cannot be found, DB2 synchronously waits until one has been freed. The system tuner should seriously consider increasing the size of the virtual buffer pool or removing some of the page sets allocated to it.

SUCCESSFUL OPEN

Number of data sets physically opened (QBSTDSO).

This counter is incremented each time a page set is physically opened.

MVS OPEN processing of a VSAM data set can be extremely expensive. The goal is to keep frequently used page sets open. While every page set must be opened at least once, the goal is to keep the heavily used page sets open as long as possible. DB2 uses a pseudo-close technique to avoid physical closes.

DB2 reduces the possible impact on recovery of large numbers of open data sets by internally shifting the R/W nature of the page set to read-only if no activity has taken place in the last five checkpoints or 60 minutes.

VP BUFFERS ALLOCATED

Number of buffers allocated for a virtual buffer pool (QBSTVPL).

This number is a snapshot of the number of pages allocated in central storage to support the virtual buffer pool.

DB2 allocates pages as needed in the DBM1 address space up to the VPSIZE parameter of that buffer pool. This number can be dynamically altered by issuing the -ALTER BUFFERPOOL command.

HP BUFFERS ALLOCATED

Number of buffers allocated for a hiperpool (QBSTHPL).

This number is a snapshot of the number of pages allocated in expanded storage to support the hiperpool.

DB2 allocates pages in expanded storage up to the HPSIZE of the buffer pool. This number can be dynamically altered by issuing the -ALTER BUFFERPOOL command.

EXPANDED STORAGE HP BUFFERS

Number of hiperpool buffers backed by expanded storage (QBSTHBE).

This number reflects the current number of buffers which have expanded storage available to back the hiperpool allocated size. However, for hiperpools defined with CASTOUT= YES (the recommended default), this number can include MVS stolen page frames if DB2 did not subsequently attempt to access the buffers.

Generally the CASTOUT parameter should be left to YES because of the severe MVS constraints which occur when expanded storage is not available for MVS paging. However, care should be used to not overallocate hiperpools.

MIGRATED DS ENCOUNTERED

Number of times migrated data sets were encountered (QBSTMIG).

This counter is incremented every time DB2 encounters a page set that has been archived and it must be migrated back.

**Tuning Tip:** In general, allowing archiving of production data is not recommended, as the cost of tape mounts and recall slow the application. It may be cost effective to allow tables to migrate in a test or System UAT environment. If this occurs frequently, consider changing the migration rules for DB2 data sets.

RECALL TIMEOUTS

Number of recall timeouts (QBSTRTO).

This counter is incremented when DB2 attempts to recover a migrated page set and must wait longer than the Recall Delay parameter (RECALLD).

If DFSMS or an equivalent product fails to recover a migrated data set, DB2 times out the recall request. The default recall delay time is 120 seconds.

HP EXP/CONTRACTION

Number of successful hiperpool expansions or contractions (QBSTHPA).

This number is incremented each time storage for a hiperpool is allocated or freed when the -ALTER BUFFERPOOL command is issued and the operation succeeds.

This is informational and is of concern only if storage cannot be obtained or released.

VP EXP/CONTRACTION

Number of successful VP expansions or contractions (QBSTVPA).

This number is incremented each time storage for a virtual buffer pool is GETMAINed or FREEMAINed when the -ALTER BUFFERPOOL command is issued and the operation succeeds.

This is informational and is of concern only if storage cannot be obtained or released.

EXPAND SOS FAIL

Number of virtual pool or hiperpool expansion failures (QBSTXFV).

This counter is incremented every time DB2 attempts to GETMAIN more central or expanded storage and fails to find memory to satisfy the request.

**Tuning Tip:** It is a serious performance concern any time this number exceeds zero. It indicates that memory is overallocated in the DBM1 address space. The performance tuner should attempt to lower buffer pool sizes (virtual pool or hiperpool) for those pools not requiring as much storage. Reducing the workload or spreading the workload among the pools can alleviate many of these problems.

HWM PREFETCH IO STREAMS

Maximum number of concurrent parallel I/O streams (QBSTXIS).

This number indicates the maximum number of I/O streams running in parallel to satisfy a query. It does not apply to work files.

**Tuning Tip:** The number is of use in sizing the VPPSEQT (virtual parallel processing sequential steal threshold) parameter for the buffer pool. The higher the number of parallel tasks, the higher the VPPSEQT parameter should be. (This is a percentage of the sequential buffers set by VPSEQT.)

PRFETCH IO STREAMS REDUCED

Number of parallel I/O streams denied—buffer shortage (QBSTJIS).

This counter is incremented when the number of buffers already reserved for sequential processing has been used and no more processes can be started. It does not apply to work files.

**Tuning Tip:** The solution to this problem is to increase the VPSEQT (virtual pool sequential steal threshold). When doing so, take care to keep a reserve for random processing. Increasing the size of the buffer pool may be a more appropriate change depending on the mix of random versus sequential processing. Another good technique is to separate tables that are frequently accessed sequentially in a different buffer pool so that the thresholds can be tuned appropriately.

REQUEST FOR PARALLELISM

Total number of requests for parallelism (QBSTPQO).

This counter is incremented each time parallel processing is requested for this virtual buffer pool, not including parallel work file support.

**Tuning Tip:** This number can assist in tuning the VPPSEQT parameter (virtual parallel processing sequential steal threshold). The more requests made, the higher this number should be.

PARALLEL REDUCTION-NO BUFF

Total number of parallel groups reduced—buffer shortage (QBSTPQF).

This number is incremented at execution time when DB2 finds enough parallel sequential buffers have not been reserved to support the parallel operation.

**Tuning Tip:** If this occurs, the VPPSEQT parameter should be increased; the VPSIZE also may need to be increased.

PREFETCH QTY CUT TO 1/2

Number of times prefetch quantity reduced to 1/2 (QBSTPL1).

This counter is incremented every time DB2 has to reduce the prefetch quantity by 50% because too many parallel tasks have reduced the amount of buffer storage available for sequential processing. For example, instead of getting 32 pages per prefetch, only 16 pages are prefetched.

**Tuning Tip:** Generally, non-zero numbers are undesirable as it indicates there are too many concurrent prefetch processes for parallel processing. This is impacting the effectiveness of prefetch by increasing I/Os. Consider reducing the VPPSEQT or increasing the VPSIZE of the buffer pool.

PREFETCH QTY CUT TO 1/4

Number of times prefetch quantity reduced to 1/4 (QBSTPL2).

This counter is incremented every time DB2 has to reduce the prefetch quantity by 75% because too many parallel tasks have reduced the amount of buffer storage available for sequential processing. For example, instead of getting 32 pages per prefetch, only 8 pages are prefetched.

**Tuning Tip:** This is a serious condition. Either reduce the VPPSEQT or increase the total VPSEQT or the VPSIZE of the buffer pool. Parallel processing should not be allowed to reduce the prefetch amount since that defeats the purpose of bringing as many pages into storage as possible.

## BP READ OPERATIONS

GETPAGES

Number of getpage requests (QBSTGET).

This counter is incremented when the Data Manager component of DB2 requests a page (hence *getpage*) from DB2's Buffer Manager. The Buffer Manager first looks into one of its virtual buffer pools, next to hiperpools (if any), and then does a read from DASD.

**Tuning Tip:** The goal in DB2 performance tuning is to keep the ratio of GETPAGEs to real I/O very high. This indicates either good sequential performance or a re-reference of random data, depending on the application.

GETPAGE SEQ REQ

Number of getpage requests for sequential access (QBSTSGT).

This is a count of getpages for all sequential access requesters including sequential prefetch, dynamic sequential prefetch, and list prefetch operations.

**Tuning Tip:** This number is useful in determining what the sequential steal threshold should be for that buffer pool. This number compared with the total number of getpages indicates the environment for random and sequential processes. Those two numbers can help to set the VPSEQT parameter.

GETPAGE RANDOM REQ

Number of getpage requests for random access (QBSTGET - QBSTSGT).

This is a count of getpages for all random access requesters.

SYNC READ I/O

Number of synchronous read I/Os (QBSTRIO).

This counter is incremented every time the DB2 Buffer Manager could not find a page in global, central, or expanded storage. DB2 must perform a physical read of DASD to obtain the necessary data. The application waits for DB2 to perform the operation.

**Tuning Tip:** Unnecessary read I/Os are one of the principal culprits in poorly tuned applications. While random I/O cannot be avoided, critical re-referenced indexes and tables can be kept in memory. At other times, the strategy is to get the data into memory before the application needs it, typically when processing the data pages sequentially. This is done with prefetch and possibly with parallel processing.

SYNC READ IO SEQ REQ

Number of synchronous I/Os for sequential access (QBSTSIO).

This count is the number of non-sequential pages found while trying to process data sequentially. DB2 must then do random reads.

**Tuning Tip:** This number can indicate data or index fragmentation. It may also indicate buffer pool thrashing. If the pool is too small, sequentially read pages may be stolen before being used. They must then be read in again.

SEQ PREFETCH REQ

Number of sequential prefetch requests (QBSTSEQ).

This counter is incremented every time a DB2 plan calls for a sequential prefetch operation, which normally attempts to bring in up to 32 pages per read I/O.

Sequential prefetch in this context includes only plans which have indicated a use of sequential prefetch in their EXPLAINs. Dynamic sequential prefetch (where DB2 decides the program is traversing the data sequentially and dynamically brings in the next 32 pages requested) is not included in this count.

SEQ PREFETCH READ IO

Number of asynchronous read I/Os because of sequential prefetch (QBSTPIO).

This counter is incremented for each physical read of data using sequential prefetch for both applications and utilities.

This is not the number of pages read; it simply represents the number of I/O operations caused by sequential processing.

SEQ PREFETCH PAGES READ

Number of asynchronous pages read by sequential prefetch (QBSTSPP).

This counter is incremented with the number of pages read using normal (not dynamic) sequential prefetch.

SEQ PREFETCH PAGES/READ

Ratio of the number of sequential prefetch pages read per sequential prefetch read I/O (QBSTSPP/QBSTPIO).

LIST PREFETCH REQUESTS

Number of list prefetch requests (QBSTLPF).

This number is incremented each time an access path requires that index keys be sorted into data order using the RID pool. List prefetch can be used with non-matching index scans and is always used when multiple indexes are used to access tables. It is also used to sort index data during a hybrid join.

At BIND time, DB2 computes whether list prefetch should be activated by estimating the number of index RIDS to be scanned and comparing that to the size of the RID pool. If the number exceeds 50%, DB2 does not activate list prefetch. At execution time, if the object would take more than 25% of the RID pool, DB2 disables list prefetch. If this occurs frequently, consider enlarging the RID pool by increasing the MAXRBLK parameter of DSNZPARM on installation panel DSNTIPC.

LIST PREFETCH READ IO

Number of asynchronous read I/Os because of list prefetch (QBSTLIO).

This is not the number of pages read; it simply represents the number of I/O operations caused by list processing.

LIST PREFETCH PAGES READ

Number of asynchronous pages read by list prefetch (QBSTLPP).

This is a count of the number of index pages read by list prefetch to satisfy non-matching index scans, multiple index access support, and certain types of join to be read into the RID pool for RID pool support.

DB2, upon sorting the RID list created by list prefetch, will then access the data using sequential prefetch to gain the performance boost of processing the data. This is an asynchronous process not charged to the calling application.

LIST PREFETCH PAGES/READ

Ratio of the number of list prefetch pages read per list prefetch read I/O (QBSTLPP/QBSTLIO).

DYNAMIC PREFETCH REQUESTS

Number of dynamic prefetch requests (QBSTDPF).

This field is incremented every time DB2 determines sequential prefetch should be dynamically activated. DB2 analyzes the data accessed to determine whether the last five of eight pages accessed are in sequential order and the application would be benefited by prefetch. DB2 then turns on sequential prefetch until the pages being accessed are no longer in sequential order.

**Tuning Tip:** Normally this assists programs by providing read-ahead buffering for processing; GETPAGEs that would have to wait for synchronous I/O now find the page in the buffer pool. This number should be monitored since the more dynamic prefetch requests activated, the more buffer pool resources can be strained. It may be necessary to alter the buffer pool size or sequential steal threshold to handle increased demands of dynamic prefetch.

DYNAMIC PREFETCH READ IO

Number of asynchronous read I/Os because of dynamic prefetch (QBSTDIO).

This is not the number of pages read; it simply represents the number of I/O operations caused by dynamic processing.

DYNAMIC PREFETCH PAGES READ

Number of asynchronous pages read by dynamic prefetch (QBSTDPP).

This number is the total number of pages accessed asynchronously using sequential prefetch because DB2 dynamically determined by sequential detection that the application was processing the pages sequentially.

**Tuning Tip:** Normally, this aids performance by having read-ahead buffering of pages in the buffer pool. System tuners and DBAs should be aware of when this happens so the number of pages allocated for sequential processing is adequate to support the workload. The VPSIZE (virtual pool size) and the VPSEQT (virtual pool sequential steal thresholds) are the numbers that should be tuned to support the workload.

DYNAMIC PREFETCH PAGES/READ

Ratio of the number of dynamic prefetch pages read per dynamic prefetch read I/O (QBSTDPP/QBSTDIO).

PF DISABLED - NO BUFFER

Number of times sequential prefetch disabled—not enough buffers (QBSTSPD).

This counter is incremented every time a virtual buffer pool reaches 90% active non-stealable buffers. Sequential prefetch, or read-ahead buffering of 32 pages per I/O, is disabled until the number of non-stealable buffers drops below the 90% mark.

**Tuning Tip:** This number should be zero or as close to zero as possible. Serious performance degradation occurs when sequential prefetch is disabled. Should it occur often, either increase the size of the virtual buffer pool or free more pages by having application code commit more often.

PREFETCH DISABLED-NO ENGINE

Number of times sequential prefetch disabled—unavailable read engine (QBSTREE).

This counter is incremented when 300 read engines are activated in a DB2 subsystem and another is needed.

**Tuning Tip:** This situation should occur rarely and the number should be close to zero. Since the 300 read engine limit is *hard coded*, the only solution available to you is to spread out the workload over a longer period of time so that the number of read engines never exceeds 300.

MVPG PAGES SYNC HP->VP

Number of synchronous reads from a hiperpool (QBSTHRE).

This number is incremented every time DB2 issues an MVS MVPG (move page) instruction to move data in a hiperpool page in expanded storage to central storage. Data being updated or reused must always be moved from expanded to central storage (virtual buffer pool).

This operation usually takes place in microseconds (millionths of seconds). The read operation is synchronous and waits until the transfer is complete. Time is charged to the calling application.

MVPG PAGES ASYNC HP->VP

Number of asynchronous pages read from a hiperpool (QBSTHRA).

This number is incremented when a sequential prefetch is issued and DB2 finds the data in a hiperpool.

The MVS MVPG (move page) instruction moves data from expanded to central storage. This operation usually takes place in microseconds (millionths of seconds). The read operation is asynchronous and time is not charged to the calling application.

HP->VP MVPG FAIL

Number of failed hiperpool reads (QBSTHRF).

This number is incremented when DB2 searches for a page in expanded storage but finds MVS has stolen the page. This can occur only if the CASTOUT parameter for that hiperpool is set to YES. DB2 must then reread the page from DASD.

**Tuning Tip:** If this number is consistently high, it indicates the hiperpool has serious competition for MVS resources. It would be appropriate to reduce the size of the hiperpool unless the application is so critical that it is deemed necessary to change CASTOUT to NO. CASTOUT NO essentially takes pages from expanded storage and makes them unavailable to MVS. This is generally not desirable as it may seriously degrade other MVS performance.

DATA MOVER ASYNC HP->VP

Number of asynchronous pages read from a hiperpool by ADMF (QBSTARA).

The Asynchronous Data Move Facility (ADMF) does mass moves of pages from expanded storage back to central storage and back. This counter is incremented with the number of pages moved by this hardware facility.

ADMF processing saves MVS MVPG instruction processing. The hardware facility can assist throughput substantially for sequential prefetch operations.

DATA MOV ASYNC FAIL HP->VP

Number of read failures with ADMF (QBSTARF).

This counter is incremented by the number of pages which have been stolen by MVS from expanded storage for MVS purposes. DB2 must then reread the data from DASD.

**Tuning Tip:** In general, if this number is high, consider reducing the size of the hiperpool since MVS should have priority. If the CASTOUT parameter is set to NO, this situation can be avoided, but this may result in serious MVS performance delays.

PAGEINS FOR READ IO

Number of page-ins required for read I/O (QBSTRPI).

This number is incremented each time a buffer page has to be read from MVS’s external paging subsystem (expanded storage or DASD) because memory is overallocated for this system.

**Tuning Tip:** Paging in buffer pools for read activity is a frequent performance concern. Buffer pools should be sized large enough to handle the asynchronous nature of write activity and read prefetch activity. However, they should not be so large that MVS will page these buffers to auxiliary storage (where the intent of fast memory transfer is defeated by the page-in operation). A rough rule of thumb is that if more than five pages per second are being paged in for read activity, there is a serious memory shortage. Either reduce the size of the buffer pool or reduce other work on the MVS image.

BPOOL HIT RATIO-ALL(%)

Buffer pool hit ratio with prefetch.

This value is calculated as

(Total Pages - Total I/O) / Total Pages x 100.0

where

Total Pages is the number of getpage requests (QBSTGET).

Total I/O is the sum of

- Synchronous read I/Os (QBSTRIO)
- Asynchronous pages read by sequential prefetch (QBSTSPP)
- Asynchronous pages read by list prefetch (QBSTLPP)
- Asynchronous pages read by dynamic prefetch (QBSTDPP)

BPOOL HIT RATIO-RANDOM(%)

Random buffer pool hit ratio.

This value is calculated as

(Sync Pages - Sync I/O) / Sync Pages x 100.0

where

Sync Pages is the number of synchronous getpage requests (QBSTGET - (QBSTSGT + QBSTNGT)).

Sync I/O is the number of synchronous read I/Os (QBSTRIO - QBSTSIO).

## BP WRITE OPERATIONS

PAGE UPDATES

Number of page updates (QBSTSWS).

This counter is incremented each time a page is updated in the pool. Once a page is updated, it is *in use* until committed or rolled back. If the same page is updated multiple times, each update intent is counted. Also, this count includes not only updates to data pages but also work file pages, so sort activity increases this count. In a data sharing environment, updated pages may cause buffer invalidation in other member DB2s.

This number reflects update activity. Ideally, DB2 writes updated pages asynchronously at a system checkpoint or through deferred write. The goal is to avoid degrading to synchronous writes.

PAGES WRITTEN

Number of pages written (QBSTPWS).

This counter is incremented with the number of updated pages externalized to DASD. The goal is to see the lowest number of pages written compared to updated pages. This indicates that multiple updates are occurring in memory, which is desirable over writes to DASD.

**Tuning Tip:** A number of factors affect this ratio including the amount of pages re-used, the buffer pool size, concurrent access, and application design. Buffer pool problems in DB2 generally tend to show up in read inefficiency. Unless page writes are totally random, tuning should be focused on read efficiency. DB2 normally handles write activity appropriately.

BUFF UPDATES/PAGES WRITTEN

Ratio of the number of page updates to the number of pages written (QBSTSWS/QBSTPWS).

SYNC WRITES

Number of immediate writes (QBSTIMW).

This counter is incremented when a buffer pool reaches 97.5% full of non-stealable buffers. DB2 then attempts to schedule all writes synchronously rather than asynchronously. This time is directly charged to all applications that are updating and seriously degrades performance.

**Tuning Tip:** This number should be zero. If non-zero, consider increasing the size of the buffer pool or ALTERing the heaviest updated page set to a different buffer pool.

ASYNC WRITE IO

Number of asynchronous write I/Os (QBSTWIO).

This number is incremented each time DB2 schedules asynchronous writes through the media manager (VSAM) to DASD. This generally occurs at system checkpoints or deferred write thresholds.

The number of asynchronous writes should far exceed any synchronous write activity.

ASYNC WRITES + SYNC WRITES

Total number of asynchronous and synchronous writes (QBSTWIO+QBSTIMW).

HORIZONTAL DEF. WR REACHED

Number of times deferred write threshold reached (QBSTDWT).

This counter is updated when there are more than 50% (DWQT default value) of updated pages in a virtual buffer pool waiting to be externalized. DB2 starts asynchronously writing the updated pages until the number drops below the threshold.

**Tuning Tip:** This can occur during heavy update periods, and by itself is not a sign of serious performance degradation. However, if this number is reached often, it could point to the need to enlarge the virtual pool or hiperpool. The 50% default is modifiable by issuing an -ALTER BUFFERPOOL command for the DWQT threshold.

VERTICAL DEFER WR REACHED

Number of times vertical deferred write threshold reached (QBSTDWV).

This counter is incremented when a single page set takes up more space than the user-defined vertical deferred write threshold (by default 10%). The goal is to prevent any single data set assigned to a buffer pool from holding on to updated buffers at the expense of the other page sets occupying the pool. Each time this occurs, DB2 writes these buffers to DASD with chained write I/O.

**Tuning Tip:** The DBA and system tuner need to know the characteristics of the buffer pool and the page sets assigned to it. If a single page set is assigned to a buffer pool, the vertical deferred write threshold can be raised. The goal is to prevent unnecessary DASD I/O caused by a too-low threshold. In general, if this threshold is reached often, the virtual buffer pool size or vertical deferred write threshold should be increased.

DM CRITICAL REACHED

Number of times Data Manager Buffer Critical reached (QBSTDMC).

This counter is incremented when a buffer pool contains 95% or more of non-stealable pages. Sequential prefetch was turned off at 90%. At 95%, DB2 parses rows instead of 4K pages. This becomes evident because more than one GETPAGE can be issued for the same page.

**Tuning Tip:** This is a sign of serious performance stress on the buffer pool. The choice of fixes are to increase the virtual pool size or to allow fewer updates during a specified time period. At 95%, the CPU overhead to read and write rows becomes extreme and is very noticeable to users.

NO WRITE ENGINE

Number of times write engine not available (QBSTWEE).

This counter is incremented when 300 write engines are activated in a DB2 subsystem and another is needed for asynchronous write I/O.

**Tuning Tip:** This situation should occur rarely and the number should be close to zero. Since the 300 write engine limit is *hard coded*, the only solution available to you is to spread out the workload over a longer period of time so that data externalization does not require more than 300 engines.

PAGES SYNC VP->HP

Number of successful synchronous page writes to a hiperpool (QBSTHWR).

This number is incremented each time DB2 moves a page from a central storage buffer pool to expanded storage. This indicates successful caching of a page.

If a hiperpool were not available, the page would be discarded. By moving the page contents to cheaper expanded storage, DB2 is increasing the effectiveness of data page reuse. Changed pages are never written to a hiperpool.

PAGES ASYNC VP->HP

Number of successful asynchronous pages written to a hiperpool (QBSTHWA).

This counter is incremented with the number of pages migrated from central to expanded storage due to a DB2 service task such as sequential prefetch.

The percent of sequentially accessed pages in the hiperpool is controlled by the HPSEQT parameter of the -ALTER BUFFERPOOL command.

PAGES WRITE FAIL VP->HP

Number of failed hiperpool writes (QBSTHWF).

This number is incremented when a shortage of hiperpool space exists and DB2 cannot move the pages from central to expanded storage. This occurs when serious MVS demands are stealing hiperpool pages.

**Tuning Tip:** When this occurs, pages are aged out from the central virtual buffer pool as they would have if the hiperpool did not exist at all. Consider reducing the size of the hiperpool so that all hiperpools can co-exist with MVS requirements.

DATA MOVER ASYNC VP->HP

Number of successful asynchronous pages written to a hiperpool by ADMF (QBSTAWA).

This counter is incremented with the number of pages moved when the Asynchronous Data Mover hardware facility moves pages from central storage to expanded storage.

Only asynchronous activities such as sequential prefetch use the ADMF facilities for this purpose.

DATA MV ASYN FAIL VP->HP

Number of write failures with ADMF (QBSTAWF).

This counter is incremented when DB2 tries to use pages in expanded storage to write data from a virtual pool and the backing storage has been stolen.

**Tuning Tip:** This situation occurs when MVS has stolen expanded storage. If this counter is consistently high, consider reducing the hiperpool size (HPSIZE) with the -ALTER BUFFERPOOL command.

PAGEINS FOR WRITE IO

Number of page-ins required for write I/O (QBSTWPI).

This reflects the number of times DB2 had to page in updated pages before writing them to DASD because these pages had been paged out to MVS auxiliary storage.

**Tuning Tip:** In general, any serious paging in a virtual pool is a sign of memory overallocation. Because of DB2’s tendency to try to write data asynchronously, paging will be less serious than a page-in required for read activity. Nevertheless, if the page-in rate is greater than roughly five pages per second, this can be a sign of serious memory overallocation.

## BP SORT/MERGE

MAX WORKFILE IN MERGE

Maximum number of work files used (QBSTWFM).

This number is the high-water mark of the work files used to support DB2 internal sort processing.

**Tuning Tip:** Ideally, each work file needs at least 16 buffers to allow DB2 to perform sequential prefetch for a single task. To tune sort, allocate as many work files as practical and isolate the sort activity from other buffer pool operations.

NUMBER MERGE PASSES

Total number of merge passes requested for sort (QBSTWFR).

This number represents the count of how many merge passes were requested from DB2. This amount varies depending on how many work files exist and how many records are being sorted.

In general, the more work files available, the more efficient DB2 internal sort will be.

MERGE PASSES/INSUFF BUFFER

Number of sort merges degraded (QBSTWFF).

This number is incremented at execution time when DB2 did not find enough buffers to support a sort operation. The number of work files used is less than requested. For DB2 to perform efficient prefetch for work files, each work file should have at least 16 dedicated buffers.

DB2 calculates current buffer availability and may then limit the maximum work files allowed, degrading the sort.

WORKFILES REJECTED LOW BUFF

Total number of work files rejected—not enough buffers (QBSTWFD).

This number is incremented when DB2 cannot find enough buffers to support all work file runs (inputs and outputs) for sort.

**Tuning Tip:** If this number is non-zero, consider moving the DSNDB07 buffer pool to a separate buffer pool with enough resources to support the maximum number of sorts performed during peak period processing.

TOTAL WORKFILES IN MERGE

Total number of work files requested for sort merge (QBSTWFT).

This number represents the optimizer’s best approximation of the number of work files needed to support each sort processed.

**Tuning Tip:** For tuning, consider the maximum number of work files requested for all sort activities during peak period processing. This will allow the best throughput for sort processing, particularly if the DSNDB07 buffers have been isolated into their own buffer pool.

WKFILES NOT CREATED-NO BUF

Number of times work files exceed buffer pool resources (QBSTMAX).

This is incremented when there is a shortage of buffer pool resources for sort activity. The limitation of memory degrades the number of work files that DB2 can use. Generally, using more work files increases the throughput of DB2 sort, much as additional SORTWK DD statements enhance the working of an external sort.

**Tuning Tip:** The best approach, if memory is available, is to dedicate a buffer pool purely for sort operations with a VPSEQT of 100% (sequential steal threshold at 100%). Since sort is a destructive read into the pool and out from the work files (in temporary work spaces in DSNDB07), isolating sort from other buffer activity can enhance the performance of sort while protecting other useful pages in different buffer pools.

PREFETCH DISABLED-WK FILES

Number of times sequential prefetch for sort disabled due to a prefetch quantity of zero (QBSTWKPD).

This counter is incremented every time DB2 determines that the number of pages which can be read with a prefetch operation during sort merge is zero. The quantity is usually 1 to 8 pages.

**Tuning Tip:** This field indicates a shortage of sequential storage in a virtual buffer pool while prefetch is active for sort merge. If this is a high number, it indicates this buffer pool needs more storage. To resolve this, increase the VPSIZE of the buffer pool, assuming memory is available. Non-zero numbers indicate serious buffer pool shortages that should be dealt with immediately.

PAGES FOR DESTRUCTIVE READ

Number of pages for which destructive read was requested (QBSTWDRP).

This number represents the high-water mark of buffer pool pages that have been used for sort operations.

DB2 internal sort uses destructive read (the previous contents of those buffers have been overwritten). This is the primary reason to allocate a separate buffer pool for DB2 sort activity.

DEQUE FROM VDWQ DISTRUCT RD

Number of pages removed from deferred write queue for sort (QBSTWBVQ).

This number represents pages no longer available for deferred write because they have been allocated for sort activity (which is destructive read).

This number of pages can be *protected* by isolating sort (DSNDB07) in its own buffer pool.

## Statistics Long Report—Page 6

The fields in this section are organized in the following categories. For the highlights, see page 254.

### DB2 COMMANDS

This section shows the number of each of these DB2 commands issued:

- ALTER BUFFERPOOL
- ARCHIVE LOG
- CANCEL THREAD
- DISPLAY ARCHIVE
- DISPLAY BUFFERPOOL
- DISPLAY DATABASE
- DISPLAY LOCATION
- DISPLAY RLIMIT
- DISPLAY THREAD
- DISPLAY TRACE
- DISPLAY UTILITY
- MOPDIFY TRACE
- RECOVER BSDS
- RECOVER INDOUBT
- RESET INDOUBT
- SET ARCHIVE
- START DATABASE
- START DB2
- START DDF
- START RLIMIT
- START TRACE
- STOP DATABASE
- STOP DB2
- STOP DDF
- STOP RLIMIT
- STOP TRACE
- TERMINATE UTILITY
- RESET GENERICLU
- ALTER GROUPBUFFERPOOL
- DISPLAY GROUPBUFFERPOOL
- DISPLAY PROCEDURE
- START PROCEDURE
- STOP PROCEDURE
- DISPLAY GROUP
- DISPLAY FUNCTION (DB2 6.1 and later)
- START FUNCTION (DB2 6.1 and later)
- STOP FUNCTION (DB2 6.1 and later)
- DISPLAY LOG (DB2 6.1 and later)
- SET LOG (DB2 6.1 and later)
- UNRECOGNIZED COMMANDS

The total number of DB2 commands issued is shown at the bottom of the column.

## DB2 API

IFI ABENDS

Count of IFI abends (QWSDSCA).

This field is incremented when an IFI trace call abends for any reason.

**Tuning Tip:** IFI abends are accompanied by a reason code on the MSTR address space log and the IFCA of the calling program. The cause should be determined and remedied.

IFI COMMAND REQUESTS

Count of IFI command requests (QWSDSCCO).

This counter is incremented when a successful IFI command is issued. Many DB2 commands can be issued through the IFI.

**Tuning Tip:** One mechanism of submitting DB2 commands (particularly trace commands) is to write a program using the IFI, as documented in the *IBM DATABASE 2 Administration Guide*.

IFI READ-A REQUESTS

Count of IFI READA requests (QWSDSCRA).

This counter is incremented when an asynchronous IFI call is made from a monitor program requesting trace data from the IFI.

**Tuning Tip:** The counter is useful in determining how many trace requests were made to DB2.

IFI READ-S REQUESTS

Count of IFI READS requests (QWSDSCRS).

This counter is incremented when synchronous IFI calls are made from a monitor program requesting trace data from the IFI.

**Tuning Tip:** This counter is useful in determining how many synchronous data requests were made to DB2.

IFI UNRECOGNIZED FUNCTIONS

Count of IFI unrecognized functions (QWSDSCU).

This field is incremented when an IFI call was made using an improper argument or parameter list.

**Tuning Tip:** The reason code from the IFCA should be examined to determine what parameter was in error using the IFI trace facility.

IFI WRITE REQUESTS

Count of IFI write requests (QWSDSCWR).

This counter is incremented when DB2 writes trace data back to a monitor program as a result of a command, READA, or READS statement.

The number of trace records written, along with their IFCIDs, can be one indicator of trace overhead.

TOTALS

Total number of IFI requests listed above.

## DATA CAPTURE

DESCRIPTOR PERFORMED

Number of data capture describes performed (QWSDCDMB).

This counter is incremented every time DB2 does catalog access to obtain the data column description to fill in the QW0185DD.

**Tuning Tip:** This process can be quite costly.

DESCRIPTOR RETURNED

Number of data capture descriptions returned (QWSDCDDD).

This is a counter of the number of descriptions returned from the IFI interface. When activated, the columns and their lengths are mapped in field QW0185DD.

This information is obtained from the catalog to allow the user to parse the row data with the data description from the catalog.

LOG READS

Number of data capture log reads of IFCID 185 (QWSDCDLR).

IFCID 185 is the specific log record used by the IFI capture facility to return before and after images of updated rows.

**Tuning Tip:** An application program must be written or a program product purchased to use this interface. It issues IFI READS requests for IFCID 185.

LOG RECORDS CAPTURED

Number of captured log records (QWSDCDLC).

This counter is incremented as log records are passed to the capture exit interface.

Any log record can be passed through either the IFI or the stand-alone data capture exit interface.

LOG RECORDS RETURNED

Number of captured log records returned (QWSDCDRR).

This is a counter DB2 maintains of log records returned through the non-IFI data capture exit.

**Tuning Tip:** This counter can be examined to determine the overhead of invoking the capture exit from the logging subsystem.

TABLE RETURNED

Number of data capture tables returned to caller (QWSDCDTB).

This counter is incremented for each table description that required a catalog lookup.

**Tuning Tip:** This permits the system administrator to determine the number of tables for which the capture facility is activated.

ROWS RETURNED

Number of data capture data rows returned (QWSDCDDR).

This counter is incremented by the number of before and after images of specific rows of data returned in IFCID 185.

IFCID 185 has a section to parse the data in the row in field QW0185DR.

## Statistics Long Report—DDF

The fields in this section are organized in the following categories.

**Note:** This page is produced only if distributed work is done.

### DRDA REMOTE LOCATIONS

REMOTE LOCATION NAME

Location name of remote site (QLSTLOCN).

Location name of the remote site involved in distributed work with this site, either as a server or requester.

Statistics for all remote locations accessed by DRDA protocol are grouped under the location name “DRDA REMOTE LOCS”. For DB2 private protocol, statistics are gathered for each remote location.

BEGIN INTERVAL DATE

Beginning date and time of the reporting interval in which DDF work was processed.

END INTERVAL DATE

Ending date and time of the reporting interval in which DDF work was processed.

TRANSACTIONS RECEIVED

Number of transactions received (QLSTTRNR).

This counter is incremented at the server location and reflects the number of threads requested by the requester.

**Tuning Tip:** This counter is useful only at the server location since DB2 maintains local site autonomy based on the number of active and inactive threads in DSNZPARM.

TRANSACTIONS SENT

Number of transactions sent (QLSTTRNS).

This counter is incremented at the requester location and reflects the total number of threads requested from the server location. Each one allocates a DBAT thread.

**Tuning Tip:** This data is useful only at the requester location since DB2 maintains local site autonomy based on the number of active and inactive threads in DSNZPARM.

CONVERSATIONS RECEIVED

Number of conversations initiated to the server (QLSTCNVR).

This counter is incremented at the server location for each conversation the server has successfully retrieved.

**Tuning Tip:** Performance tuning is the art of getting as many data rows possible in as few conversations as possible.

CONVERSATIONS SENT

Number of conversations initiated from the requester location (QLSTCNVS).

This counter is incremented at the requester location and reflects the number of specific conversations sent to all server locations.

Conversations are specific instances of transmitting data back and forth. Sessions are logical connections between requesters and servers. It is analogous to a telephone call in which the call is a session and each party talking is a conversation.

CONVERSATIONS QUEUED

Number of conversations queued (QLSTCNVQ).

This counter is incremented at the requester location every time a conversation request to VTAM is initiated but cannot yet be started.

**Tuning Tip:** The way to avoid queuing conversational requests from a DB2 is to increase the conversation limits column in the SYSIBM.SYSLUMODEs table to a larger number. This should be done in conjunction with the VTAM system programmer to determine if the network can handle increased numbers of conversations.

SQL STATEMENTS RECEIVED

Number of SQL statements received from the requester (QLSTSQLR).

This counter is incremented at the server location when an SQL statement is received.

**Tuning Tip:** This information, along with other message information, can help determine the relative efficiency of the distributed applications and can, in conjunction with other data items, help in the network tuning of distributed applications.

SQL STATEMENTS SENT

Number of SQL statements sent to the remote server (QLSTSQLS).

This counter is incremented at the requester location statistics when an SQL statement is sent.

**Tuning Tip:** This information, along with other message information, can help determine the relative efficiency of the distributed applications and can, in conjunction with other data items, help in the network tuning of distributed applications.

1PH COMMITS RECEIVED

Number of commit requests received from the requester (QLSTCOMR).

This counter is incremented at the server location for single phase commits only.

Two-phase commit operations (updating distributed transactions, CICS, and IMS) are not included.

1PH COMMITS SENT

Number of commit requests sent to the server (QLSTCOMS).

This counter is incremented at the requester and reflects the number of single phase commits sent (TSO, batch, CAF).

Two-phase commit operations (updating distributed transactions, CICS, and IMS) are not included.

1PH ROLLBACK RECEIVED

Number of rollback requests received from the requester (QLSTABRR).

This counter is incremented at the server location reflecting the number of single phase rollback requests received.

Two-phase commit operations (updating distributed transactions, CICS, and IMS) are not included.

1PH ROLLBACK SENT

Number of rollback requests sent to the server (QLSTABRS).

This counter is incremented at the requester location and reflects the number of single phase rollback requests.

Two-phase commit operations (updating distributed transactions, CICS, and IMS) are not included.

ROWS RECEIVED

Number of rows received from the server (QLSTROWR).

This counter is incremented at the requester and reflects the number of rows retrieved from the server.

**Tuning Tip:** From a performance perspective, the ideal is to retrieve as many rows in as few conversational messages as possible. This is accomplished using a feature of DB2 called block fetch. Block fetch is used when a cursor is determined unambiguously to be a read-only cursor. The use of the FOR FETCH ONLY clause along with the bind option of CURRENTDATA(NO) can help DB2 achieve this goal. Block fetch is always most effective with system-directed SQL (three-part names without a CONNECT statement, sometimes called DB2 private protocols). In using private protocols, DB2 fits as many rows as possible into a single 32K message. The typical overhead in distributed performance is network bandwidth. If the application can send fewer messages, the distributed application is more efficient.

ROWS SENT

Number of rows sent to the requester (QLSTROWS).

This counter is incremented at the server for each row sent to a remote requester.

This number, combined with the number of messages (conversations), can be used in determining whether efficiency techniques such as block fetch are being used.

MESSAGES RECEIVED

Number of messages received from the remote location (QLSTMSGR).

This counter is incremented at the receiving location reflecting the number of messages this location received.

The count of messages received can often be more than the number of messages sent because of the way distributed SQL statements are processed.

MESSAGES SENT

Number of messages sent to the remote location (QLSTMSGS).

This counter is incremented at the originating location. It reflects a count of each VTAM message.

A message, in VTAM terms is a group of characters and control bit sequences which are transmitted as a single unit of data, including header information and data.

BYTES RECEIVED

Number of bytes received from the server (QLSTBYTR).

This counter is incremented at the requester reflecting the number of bytes retrieved by the server.

The number of bytes received may be greater than the number sent since the overhead of network transmission tends to add message information in the form of additional VTAM header blocks.

BYTES SENT

Number of bytes sent to the requester (QLSTBYTS).

This count represents the number of bytes sent by the server. This counter is incremented at the server location.

**Tuning Tip:** The number of bytes per message can help the network tuner decide the appropriate RUSIZE and pacing to fit the network.

BLOCKS RECEIVED - BLK FETCH

Number of blocks received using block fetch (QLSTBRBF).

This counter, incremented at the requester location, indicates how many blocks were received to process remote requesters.

**Tuning Tip:** Each block costs VTAM transmission time. The more rows that can be transmitted per block, the better performance for distributed transactions will be.

BLOCKS SENT - BLK FETCH

Number of blocks transmitted using block fetch (QLSTBTBF).

This counter, incremented at the server location, indicates how many blocks were sent to remote requesters.

**Tuning Tip:** Each block costs VTAM transmission time. The more rows that can be transmitted per block, the better performance for distributed transactions will be.

MESSAGES IN BUFFER

Number of rows transmitted or received using block fetch (QLSTBROW).

This counter, incremented at both requester and server, indicates the number of rows transmitted or received using a block fetch operation.

**Tuning Tip:** Generally speaking, the higher the number of rows per block, the better performance for distributed transactions will be.

SWITCH LIMITED BLK PROTOCOL

Number of switches from continuous to limited block fetch (QLSTCBLB).

This counter is incremented at the requester location and applies only to system-directed applications (three-part names, no CONNECT statement, sometimes called private protocols). It reflects the number of times asynchronous block fetch had to be reduced to a more limited form of block fetch because of a lack of network resources.

This situation occurs when too many conversations are occurring at this site for VTAM to handle the asynchronous fetch of data. It needs to be resolved by the MVS and VTAM systems programmers to allow more conversations to take place.

SQL BOUND FOR REMOTE ACCESS

Number of SQL statements bound for remote access (QLSTRBND).

This value is incremented at the requester location each time a system-directed SQL statement is encountered (three part name, no CONNECT statement, DB2 private protocol), where dynamic requests occur at the server.

One of the expensive features of the DB2 private protocol is that the SQL statement is dynamically run on the server site and thus incurs extra overhead at the server.

PREPARE REQUEST RECEIVED

Number of prepare requests received by the participant (QLSTPRRC).

This counter is incremented by the participant (usually the requester) each time the coordinator issues the prepare to all participants in DB2 two-phase commit protocols.

Two-phase commit in distributed terms works similarly to two-phase commit on a local system from CICS or IMS. A coordinator (in this case the server) tells all participating DB2 subsystems to issue a phase 1 prepare to commit. The transaction completes when all participants indicate they have logged all data and phase 2 has been accomplished.

PREPARE REQUEST SENT

Number of prepare requests sent to the participant (QLSTPRSE).

This counter is incremented by the coordinator to all participants in DB2 two-phase commit protocols when the coordinator requests all participants to log data and complete phase 1 of the commit operation.

Two-phase commit in distributed terms works similarly to two-phase commit on a local system from CICS or IMS. A coordinator (in this case the server) tells all participating DB2 subsystems to issue a phase 1 prepare to commit. The transaction completes when all participants indicate they have logged all data and phase 2 has been accomplished.

LAST AGENT REQUEST RECEIVED

Number of last agent requests received (QLSTLARC).

When DB2 is the coordinator to another DB2, this counter is incremented by the coordinator each time a) a conversation is deallocated and b) the conversation was not used since the last commit. This applies only to two-phase commit processing. The advantage is that this permits DB2 to save message overhead to acknowledge the commit request. If the application program issues a RELEASE SQL verb BEFORE the commit, another message can be saved.

**Tuning Tip:** The art of VTAM distributed performance tuning is to reduce the number of messages sent or received. By taking advantage of last agent requests and sensible use of the RELEASE SQL verb, the application program can help save message traffic across a distributed transaction.

LAST AGENT REQUEST SENT

Number of last agent requests sent to the coordinator (QLSTLASE).

When DB2 is the requester to another DB2, this counter is incremented by the participant each time a) a conversation is deallocated and b) the conversation was not used since the last commit. This applies only to two-phase commit processing. The advantage is that this permits DB2 to save message overhead to acknowledge the commit request. If the application program issues a RELEASE SQL verb BEFORE the commit, another message can be saved.

**Tuning Tip:** The art of VTAM distributed performance tuning is to reduce the number of messages sent or received. By taking advantage of last agent requests and sensible use of the RELEASE SQL verb, the application program can help save message traffic across a distributed transaction.

2PH COMMIT REQUEST RECEIVED

Number of two-phase commits received from the coordinator (QLSTCRRC).

This counter is incremented by the participant each time a phase 1 request is received.

In a two-phase commit operation the coordinator then sends a phase 2 request to all participants involved in an update. Assuming all vote YES, the participants and coordinator release locks and issue a *forget* message to all logs.

2PH COMMIT REQUEST SENT

Number of two-phase commits sent to the participant (QLSTCRSE).

This counter is incremented by the coordinator each time a phase 1 request is sent.

In a two-phase commit operation the coordinator sends a phase 2 request to all participants involved in an update. Assuming all vote YES, the participants and coordinator release locks and issue a *forget* message to all logs.

2PH BACKOUT REQ RECEIVED

Number of backout requests received by the participant (QLSTBKRC).

This counter is incremented when the participating DB2 receives a backout request from the coordinator. This usually occurs when the coordinator has received a NO vote from phase 1 processing or an abend has occurred.

Backout processing causes all participants and coordinator to backout to the last commit.

2PH BACKOUT REQ SENT

Number of backout requests sent to the participant (QLSTBKSE).

This counter is incremented by the coordinator when a NO vote has been received at end of phase 1 or a transaction has abended.

Backout processing causes all participants and coordinator to backout to the last commit.

2PH FORGET RESP RECEIVED

Number of forget responses received by the participant (QLSTRRRC).

This situation occurs and a counter incremented by the coordinator in a two-phase commit operation where any participant has not applied any updates.

The forget response takes the participant out of the voting chain when requested to prepare for commit.

2PH FORGET RESP SENT

Number of forget responses sent to the coordinator (QLSTRRSE).

This situation occurs and a counter incremented by the participant in a two-phase commit operation when the participant has not applied any updates.

The forget response takes the participant out of the voting chain when requested to prepare for commit.

2PH COMMIT RESP RECEIVED

Number of request commits received by the coordinator (QLSTVYRC).

This counter is incremented by a coordinator when it has received *request commits* from all participants involved in the update at the end of phase 1.

This counter is incremented when the participant has been involved in an update; otherwise a forget record would have been received.

2PH COMMIT RESP SENT

Number of request commits sent to the coordinator (QLSTVYSE).

This counter is incremented by a participant when it has completed phase 1 processing and is ready to perform phase 2 processing.

This counter is incremented only when the participant has been involved in an update in this unit of work.

2PH BACKOUT RESP RECEIVED

Number of backout requests received by the coordinator (QLSTVNRC).

This counter is incremented by a coordinator in a two-phase commit operation that has one or more participants voting NO.

Normal processing after this occurs is that the coordinator issues a backout request to all participants.

2PH BACKOUT RESP SENT

Number of backout requests sent to the coordinator (QLSTVNSE).

This counter is incremented by a participant in a two-phase commit operation that has voted NO because something has failed during phase 1 processing.

NO votes usually correlate to abends at the requester.

REMOTE INDOUBT THD

Number of threads that became indoubt (QLSTINDT).

An indoubt situation occurs when a failure occurs after a successful prepare but before a successful commit.

**Tuning Tip:** Indoubt threads are usually resolved automatically upon successful restart of the failing participant or coordinator and the network path used to send the messages. The problem occurs when the systems are intact but communication has been severed. This can cause indoubt threads to be maintained for long periods of time. A -RESOLVE INDOUBT command may be issued to free these locks but ***only*** after a responsible system programmer has determined the correct course of action: COMMIT or ABORT.

REMOTE COMMIT

Number of commits performed with remote location as the coordinator (QLSTCPTR).

This counter is incremented when a participant successfully completes the phase 2 process.

This indicates that two-phase processing has resulted in a successful commit.

REMOTE ROLLBACK

Number of rollbacks performed with remote location as the coordinator (QLSTRBTR).

This counter is incremented when a participant has received a ROLLBACK request from the coordinator.

**Tuning Tip:** This indicates that one or more participants or coordinator received a NO vote to commit. High numbers indicate problems with the application on one or more sites.

# Part 3. Reports from DB2 Tables

This part describes each of the performance reports produced from DB2 tables as follows:

- A sample report
- Definitions for each field in the report

# Chapter 11. Introduction

The performance reports from DB2 tables are organized into three groups:

- Accounting data
- Statistics data
- Audit data

This chapter provides an introduction to these reports. It discusses the report naming conventions and provides a table of all the reports grouped by DB2 area of activity (see Table 2 on page 336).

A report name is constructed as

```
xxzzzzzy
```

where

xx Data source

| | |
|---|---|
| AC | Detailed accounting data |
| SA | Summarized accounting data |
| ST | Statistics data |
| AU | Audit data |

zzzzz Report name

For buffer pool or global buffer pool reports, these characters are further refined as:

```
BFpzz (buffer pool), or GBpzz (global buffer pool)
```

where

p Groups the buffer pools by:

| | |
|---|---|
| A | All buffer pools |
| X | Single buffer pool |

zz Defines

| | |
|---|---|
| T | Totals |
| A | Averages |
| VP | Virtual pool activity |
| PF | Prefetch activity |
| HP | Hiperpool activity |
| EX | Exceptions (buffer pool or global pool) |
| null | Global pool activity |
| 2 | Extended report |

y Report orientation

| | |
|---|---|
| P | Plan oriented report (from accounting data) |
| A | AUTHID oriented report (from accounting data) |
| null | Date oriented report (from statistics data) or report oriented by both AUTHID and plan (from audit data) |

Table 2. DB2 Performance Reports

| Report | Title | Page | Area |
|---|---|---|---|
| xxOVERy | DB2 Accounting Overview Report<br>• ACOVERA—Detail Accounting, by AUTHID<br>• ACOVERP—Detail Accounting, by Plan<br>• SAOVERA—Summary Accounting, by AUTHID<br>• SAOVERP—Summary Accounting, by Plan | 346 | General DB2 System |
| xxOVDFy | DB2 DDF Accounting Overview Report<br>• ACOVDFA—DDF Detail Accounting, by AUTHID and Location<br>• ACOVDFP—DDF Detail Accounting, by Plan and Location<br>• SAOVDFA—DDF Summary Accounting, by AUTHID and Location<br>• SAOVDFP—DDF Summary Accounting, by Plan and Location | 348 | |
| STOVER | DB2 Statistics Overview Report—Averages | 385 | |
| STOVERT | DB2 Statistics Overview Report—Totals | 387 | |
| ACEXCEP | DB2 Accounting Exceptions Report | 350 | |
| STEXTHD | DB2 Statistics Thread-Related Exceptions Report | 389 | |
| STEXSYS | DB2 Statistics System Exceptions Report | 391 | |
| xxSQLy | DB2 Accounting SQL Report—Averages<br>• ACSQLA—Detail Accounting, by AUTHID<br>• ACSQLP—Detail Accounting, by Plan<br>• SASQLA—Summary Accounting, by AUTHID<br>• SASQLP—Summary Accounting, by Plan | 352 | User Activity |
| STSQL | DB2 Statistics SQL Report—Averages | 392 | |
| xxDYNCy | DB2 Accounting Dynamic SQL Statement Cache Report<br>• ACDYNCA—Detail Accounting, by AUTHID<br>• ACDYNCP—Detail Accounting, by Plan<br>• SADYNCA—Summary Accounting, by AUTHID<br>• SADYNCP—Summary Accounting, by Plan | 354 | |
| STDYNC | DB2 Statistics Dynamic SQL Statement Cache Report | 394 | |
| xxCPUy | DB2 Accounting CPU/Elapsed Time Report<br>• ACCPUA—Detail Accounting, by AUTHID<br>• ACCPUP—Detail Accounting, by Plan<br>• SACPUA—Summary Accounting, by AUTHID<br>• SACPUP—Summary Accounting, by Plan | 355 | |

Table 2. DB2 Performance Reports (Continued)

| Report | Title | Page | Area |
|---|---|---|---|
| xxCPDFy | DB2 DDF Accounting CPU/Elapsed Time Report<br>• ACCPDFA—DDF Detail Accounting, by AUTHID<br>• ACCPDFP—DDF Detail Accounting, by Plan<br>• SACPDFA—DDF Summary Accounting, by AUTHID<br>• SACPDFP—DDF Summary Accounting, by Plan | 357 | User Activity (continued) |
| xxPKGy | DB2 Accounting Package Report<br>• ACPKGA—Package Accounting, by AUTHID<br>• ACPKGP—Package Accounting, by Plan<br>• SAPKGA—Package Summary Accounting, by AUTHID<br>• SAPKGP—Package Summary Accounting, by Plan | 358 | |
| xxPKGS | DB2 Summary Accounting Package Suspensions Report<br>• ACPKGS—Accounting Package Suspensions<br>• SAPKGS—Package Summary Accounting Suspensions | 359 | |
| STCPU | DB2 Statistics CPU Time Report | 396 | |
| xxPLIOy | DB2 Accounting Parallelism Report<br>• ACPLIOA—Detail Accounting, by AUTHID<br>• ACPLIOP—Detail Accounting, by Plan<br>• SAPLIOA—Summary Accounting, by AUTHID<br>• SAPLIOP—Summary Accounting, by Plan | 361 | |
| STPLIO | DB2 Statistics Parallelism Report | 397 | |
| xxSPRCy | DB2 Accounting Stored Procedures Report<br>• ACSPRCA—Detail Accounting, by AUTHID<br>• ACSPRCP—Detail Accounting, by Plan<br>• SASPRCA—Summary Accounting, by AUTHID<br>• SASPRCP—Summary Accounting, by Plan | 363 | |
| STSPROC | DB2 Statistics Stored Procedures Report | 399 | |

Table 2. DB2 Performance Reports (Continued)

| Report | Title | Page | Area |
|---|---|---|---|
| xxDDFxy | DB2 Accounting DDF Local Report<br>• ACDDFLA—Local Activity Detail DDF Accounting, by AUTHID<br>• ACDDFLP—Local Activity Detail DDF Accounting, by Plan<br>• SADDFLA—Local Activity Summary DDF Accounting, by AUTHID<br>• SADDFLP—Local Activity Summary DDF Accounting, by Plan | 365 | DDF Activity |
| | DB2 Accounting DDF Remote Report<br>• ACDDFRA—Remote Activity Detail DDF Accounting, by AUTHID<br>• ACDDFRP—Remote Activity Detail DDF Accounting, by Plan<br>• SADDFRA—Remote Activity Summary DDF Accounting, by AUTHID<br>• SADDFRP—Remote Activity Summary DDF Accounting, by Plan | 367 | |
| STOVDFT | DB2 Statistics DDF Overview Report—Totals | 401 | |
| xxLOCKy | DB2 Accounting Lock/Latch Report<br>• ACLOCKA—Detail Accounting, by AUTHID<br>• ACLOCKP—Detail Accounting, by Plan<br>• SALOCKA—Summary Accounting, by AUTHID<br>• SALOCKP—Summary Accounting, by Plan | 369 | Locks |
| STLOCK | DB2 Statistics Lock Report | 402 | |
| xxGLOKy | DB2 Accounting Global Lock Report<br>• ACGLOKA—Detail Accounting, by AUTHID<br>• ACGLOKP—Detail Accounting, by Plan<br>• SAGLOKA—Summary Accounting, by AUTHID<br>• SAGLOKP—Summary Accounting, by Plan | 371 | |
| STGLOCK | DB2 Statistics Global Lock Report | 404 | |
| STEDM | DB2 Statistics EDM Pool Report | 406 | EDM Pool |
| STEDM2 | DB2 Statistics EDM Pool Extended Report | 408 | |

Table 2. DB2 Performance Reports (Continued)

| Report | Title | Page | Area |
|---|---|---|---|
| xxBFxTy | DB2 Accounting Buffer Pool Report—Totals<br>• ACBFATA—Detail Accounting, by AUTHID—All Buffer Pools<br>• ACBFATP—Detail Accounting, by Plan—All Buffer Pools<br>• SABFATA—Summary Accounting, by AUTHID—All Buffer Pools<br>• SABFATP—Summary Accounting, by Plan—All Buffer Pools<br>• ACBFXTA—Detail Accounting, by AUTHID—Single Buffer Pool<br>• ACBFXTP—Detail Accounting, by Plan—Single Buffer Pool<br>• SABFXTA—Summary Accounting, by AUTHID—Single Buffer Pool<br>• SABFXTP—Summary Accounting, by Plan—Single Buffer Pool | 373 | Buffer Pools |
| xxBFxAy | DB2 Accounting Buffer Pool Report—Averages<br>• ACBFAAA—Detail Accounting, by AUTHID—All Buffer Pools<br>• ACBFAAP—Detail Accounting, by Plan—All Buffer Pools<br>• SABFAAA—Summary Accounting, by AUTHID—All Buffer Pools<br>• SABFAAP—Summary Accounting, by Plan—All Buffer Pools<br>• ACBFXAA—Detail Accounting, by AUTHID—Single Buffer Pool<br>• ACBFXAP—Detail Accounting, by Plan—Single Buffer Pool<br>• SABFXAA—Summary Accounting, by AUTHID—Single Buffer Pool<br>• SABFXAP—Summary Accounting, by Plan—Single Buffer Pool | 375 | |
| STBFpVP | DB2 Statistics Buffer Pool Activity Report<br>• STBFAVP—All Buffer Pools<br>• STBFXVP—Single Buffer Pool | 409 | |
| STBFpPF | DB2 Statistics Buffer Pool Prefetch Activity Report<br>• STBFAPF—All Buffer Pools<br>• STBFXPF—Single Buffer Pool | 411 | |
| STBFpHP | DB2 Statistics Hiperpool Activity Report<br>• STBFAHP—All Buffer Pools<br>• STBFXHP—Single Buffer Pool | 413 | |
| STBFpEX | DB2 Statistics Buffer Pool Exceptions Report<br>• STBFAEX—All Buffer Pools<br>• STBFXEX—Single Buffer Pool | 415 | |

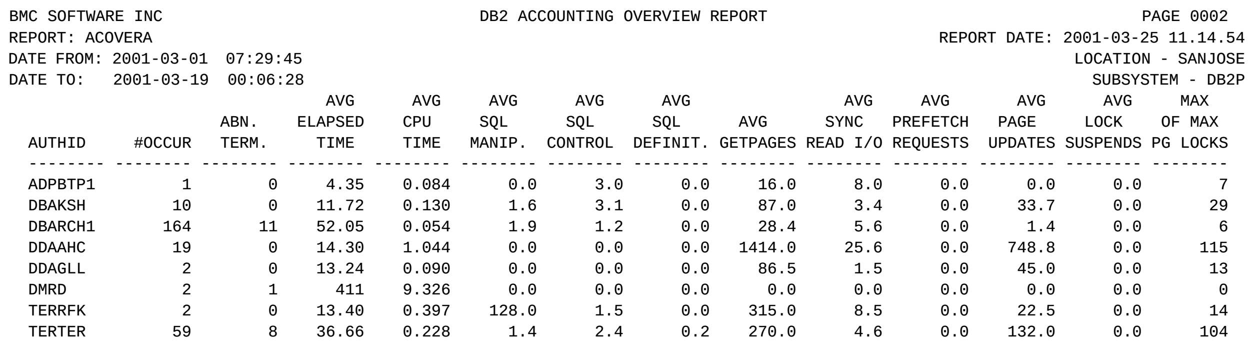
Table 2. DB2 Performance Reports (Continued)

| Report | Title | Page | Area |
|---|---|---|---|
| xxRIDy | DB2 Accounting RID List Access Report<br>• ACRIDA—Detail Accounting, by AUTHID<br>• ACRIDP—Detail Accounting, by Plan<br>• SARIDA—Summary Accounting, by AUTHID<br>• SARIDP—Summary Accounting, by Plan | 377 | Buffer Pools (Continued) |
| STRID | DB2 Statistics RID Activity Report | 417 | |
| xxGBATy | DB2 Accounting Global BPool Report—Totals<br>• ACGBATA—Detail Accounting, by AUTHID<br>• ACGBATP—Detail Accounting, by Plan<br>• SAGBATA—Summary Accounting, by AUTHID<br>• SAGBATP—Summary Accounting, by Plan | 378 | |
| xxGBAAy | DB2 Accounting Global BPool Report—Averages<br>• ACGBAAA—Detail Accounting, by AUTHID<br>• ACGBAAP—Detail Accounting, by Plan<br>• SAGBAAA—Summary Accounting, by AUTHID<br>• SAGBAAP—Summary Accounting, by Plan | 380 | |
| STGBp | DB2 Statistics Global BPool Activity Report<br>• STGBA—All Buffer Pools<br>• STGBX—Single Buffer Pool | 418 | |
| STGBp2 | DB2 Statistics Global BPool Activity Extended Report<br>• STGBA2—All Buffer Pools<br>• STGBX2—Single Buffer Pool | 420 | |
| STGBpEX | DB2 Statistics Global BPool Exceptions Report<br>• STGBAEX—All Buffer Pools<br>• STGBXEX—Single Buffer Pool | 422 | |
| STLOG | DB2 Statistics Logging Report | 424 | Logs |
| AUSUM | DB2 Audit Summary Report | 428 | Audit |
| AUDML | DB2 Audited DML Access Report | 435 | |
| AUFAIL | DB2 Authorization Failures Report | 430 | |
| AUDGRV | DB2 Authorization Control—GRANTs / REVOKEs Report | 431 | |
| AUDDL | DB2 Audited DDL Access Report | 433 | |
| AUDMLB | DB2 DML at BIND Report | 436 | |
| AUCHNG | DB2 AUTHID Change Report | 438 | |
| AUUTIL | DB2 Audit Utility Access Report | 440 | |
| AUDTL | DB2 Audit Detail Report | 442 | |

# Chapter 12.  Report Format

The reports from DB2 tables provided by Performance Reporter contain a common format for the report page heading, as described in this chapter. The QMF-produced reports are very similar to those produced by the batch program DPRREPT; however, because of limitations within QMF, some differences are noted.

The general page format is shown in Figure 30.



```
BMC SOFTWARE INC                                DB2 ACCOUNTING OVERVIEW REPORT                                  PAGE 0002
REPORT: ACOVERA                                                                                  REPORT DATE: 2001-03-25 11.14.54
DATE FROM: 2001-03-01  07:29:45                                                                                LOCATION - SANJOSE
DATE TO:   2001-03-19  00:06:28                                                                                SUBSYSTEM - DB2P
                                   AVG      AVG      AVG      AVG      AVG               AVG      AVG      AVG      AVG      MAX
                          ABN.  ELAPSED     CPU     SQL      SQL      SQL      AVG      SYNC  PREFETCH   PAGE     LOCK    OF MAX
  AUTHID     #OCCUR     TERM.     TIME     TIME   MANIP.  CONTROL  DEFINIT. GETPAGES READ I/O REQUESTS  UPDATES SUSPENDS PG LOCKS
  -------- -------- -------- -------- -------- -------- -------- -------- -------- -------- -------- -------- -------- --------
  ADPBTP1         1        0     4.35    0.084      0.0      3.0      0.0     16.0      8.0      0.0      0.0      0.0        7
  DBAKSH         10        0    11.72    0.130      1.6      3.1      0.0     87.0      3.4      0.0     33.7      0.0       29
  DBARCH1       164       11    52.05    0.054      1.9      1.2      0.0     28.4      5.6      0.0      1.4      0.0        6
  DDAAHC         19        0    14.30    1.044      0.0      0.0      0.0   1414.0     25.6      0.0    748.8      0.0      115
  DDAGLL          2        0    13.24    0.090      0.0      0.0      0.0     86.5      1.5      0.0     45.0      0.0       13
  DMRD            2        1      411    9.326      0.0      0.0      0.0      0.0      0.0      0.0      0.0      0.0        0
  TERRFK          2        0    13.40    0.397    128.0      1.5      0.0    315.0      8.5      0.0     22.5      0.0       14
  TERTER         59        8    36.66    0.228      1.4      2.4      0.2    270.0      4.6      0.0    132.0      0.0      104
```

Figure 30.  Sample Performance Reporter Report Page Layout

The headings shown on each of the reports are as follows:

copyright

Company name denotes the copyright of the company.

Positioned on left side of line 1 of report page.

title

Report title that describes the nature of the data contained within this report.

Positioned in center of line 1 of report page.

PAGE

Page number of this report. On reports produced by DPRREPT page numbering starts at 1 for each report selected within a run of this program.

Positioned on right side of line 1 of report page.

REPORT

Name of the query that produced this report. It is either a QMF query name or the BBPARM member containing the SQL query processed by DPRREPT.

Positioned on left of line 2 of report page.

REPORT DATE

Date this report was run or produced.

Positioned on right side of line 2 of report page.

DATE FROM

Start datetime range of data encountered when producing this report. This field is a report footer when QMF is used. If the SQL SELECT statement is not programmed according to the restrictions noted in "Customizing Reports" on page 83, blanks are printed by the DPRREPT program. All MAINVIEW for DB2 reports provide this information.

Positioned on left side of line 3 of report page.

LOCATION

Name of the location producing data contained in the following report pages. A page break occurs when this field changes. If the SQL SELECT statement is not programmed according to the restrictions noted in "Customizing Reports" on page 83, this field is not printed by the DPRREPT program. All MAINVIEW for DB2 reports contain this information.

Positioned on right side of line 3 of report page.

DATE TO

Ending datetime range of data encountered when producing this report. This field is a report footer when QMF is used. If the SQL SELECT statement is not programmed according to the restrictions noted in "Customizing Reports" on page 83, blanks are printed by the DPRREPT program. All MAINVIEW for DB2 reports provide this information.

Positioned on left side of line 4 of report page.

SUBSYSTEM

DB2 subsystem producing data contained in the following report pages. A page break occurs when this field changes. If the SQL SELECT statement is not programmed according to the restrictions noted in "Customizing Reports" on page 83, this field is not be printed by the DPRREPT program. All MAINVIEW for DB2 reports provide this information.

Positioned on right side of line 4 of report page.

# Chapter 13. Accounting Reports

This chapter describes the accounting reports produced by Performance Reporter, which include

- General DB2 system reports
  - The DB2 Accounting Overview Report
    - ACOVERA
    - ACOVERP
    - SAOVERA
    - SAOVERP
  - The DB2 DDF Accounting Overview Report
    - ACOVDFA
    - ACOVDFP
    - SAOVDFA
    - SAOVDFP
  - The DB2 Accounting Exceptions Report
    - ACEXCEP
- User activity reports
  - The DB2 Accounting SQL Report—Averages
    - ACSQLA
    - ACSQLP
    - SASQLA
    - SASQLP
  - The DB2 Accounting Dynamic SQL Statement Cache Report
    - ACDYNCA
    - ACDYNCP
    - SADYNCA
    - SADYNCP
  - The DB2 Accounting CPU/Elapsed Time Report
    - ACCPUA
    - ACCPUP
    - SACPUA
    - SACPUP
  - The DB2 Accounting DDF CPU/Elapsed Time Report
    - ACCPDFA
    - ACCPDFP
    - SACPDFA
    - SACPDFP

- The DB2 Accounting Package Report
  - ACPKGA
  - ACPKGP
  - ACPKGS
  - SAPKGA
  - SAPKGP
  - SAPKGS
- The DB2 Accounting Parallelism Report
  - ACPLIOA
  - ACPLIOP
  - SAPLIOA
  - SAPLIOP
- The DB2 Accounting Stored Procedures Report
  - ACSPRCA
  - ACSPRCP
  - SASPRCA
  - SASPRCP
- The DB2 Local DDF Accounting
  - ACDDFLA
  - ACDDFLP
  - SADDFLA
  - SADDFLP
- The DB2 Remote DDF Accounting
  - ACDDFRA
  - ACDDFRP
  - SADDFRA
  - SADDFRP

- Lock reports
  - The DB2 Accounting Lock/Latch Report
    - ACLOCKA
    - ACLOCKP
    - SALOCKA
    - SALOCKP
  - The DB2 Accounting Global Lock Report
    - ACGLOKA
    - ACGLOKP
    - SAGLOKA
    - SAGLOKP

- Buffer Pool reports
  - The DB2 Accounting Buffer Pool Report—Totals
    - ACBFATA
    - ACBFATP
    - SABFATA
    - SABFATP
    - ACBFXTA
    - ACBFXTP
    - SABFXTA
    - SABFXTP
  - The DB2 Accounting Buffer Pool Report—Averages
    - ACBFAAA
    - ACBFAAP
    - SABFAAA
    - SABFAAP
    - ACBFXAA
    - ACBFXAP
    - SABFXAA
    - SABFXAP
  - The DB2 Accounting RID List Access Report
    - ACRIDA
    - ACRIDP
    - SARIDA
    - SARIDP
  - The DB2 Accounting Global BPool Report—Totals
    - ACGBATA
    - ACGBATP
    - SAGBATA
    - SAGBATP
  - The DB2 Accounting Global BPool Report—Averages
    - ACGBAAA
    - ACGBAAP
    - SAGBAAA
    - SAGBAAP

# General DB2 System Reports

The general DB2 system reports provide detailed or summary accounting information about the amount of work by user or plan, the DDF activity, or abnormal conditions occurring within the DB2 system.

## xxOVERy—DB2 Accounting Overview Report

The DB2 Accounting Overview Report provides an overview of the amount of work for which each user or plan is responsible within the DB2 system. It is produced from either detailed accounting data (ACOVERA and ACOVERP) or summary accounting data (SAOVERA and SAOVERP). It can be plan-oriented (ACOVERP and SAOVERP) or user-oriented (ACOVERA and SAOVERA).

```
BMC SOFTWARE INC                               DB2 ACCOUNTING OVERVIEW REPORT                                          PAGE 0002
REPORT: ACOVERA                                                                                 REPORT DATE: 2001-03-25 11.14.54
DATE FROM: 2001-03-01 07:29:45                                                                                  LOCATION - SANJOSE
DATE TO:   2001-03-19 00:06:28                                                                                 SUBSYSTEM - DB2P

                              AVG      AVG     AVG      AVG       AVG               AVG       AVG      AVG      AVG       MAX
                    ABN.    ELAPSED    CPU     SQL      SQL       SQL       AVG     SYNC    PREFETCH   PAGE     LOCK     OF MAX
AUTHID    #OCCUR    TERM.     TIME     TIME    MANIP.    CONTROL   DEFINIT. GETPAGES READ I/O REQUESTS  UPDATES SUPSPENDS PG LOCKS
-------- ------- --------- -------- ------- -------- --------- --------- -------- -------- --------- ------- --------- --------
ADPBTP1        1         0     4.35   0.084      0.0       3.0       0.0      16.0      8.0      0.0      0.0       0.0       7
DBAKSH        10         0    11.72   0.130      1.6       3.1       0.0      87.0      3.4      0.0     33.7       0.0      29
DBARCH1      164        11    52.05   0.054      1.9       1.2       0.0      28.4      5.6      0.0      1.4       0.0       6
DDAAHC        19         0    14.30   1.044      0.0       0.0       0.0    1414.0     25.6      0.0    748.8       0.0     115
DDAGLL         2         0    13.24   0.090      0.0       0.0       0.0      86.5      1.5      0.0     45.0       0.0      13
DMRD           2         1      411   9.326      0.0       0.0       0.0       0.0      0.0      0.0      0.0       0.0       0
TERRFK         2         0    13.40   0.397    128.0       1.5       0.0     315.0      8.5      0.0     22.5       0.0      14
TERTER        59         8    36.66   0.228      1.4       2.4       0.2     270.0      4.6      0.0    132.0       0.0     104
```

Figure 31.  DB2 Accounting Overview Report

AUTHID
: Authorization ID of the user accessing this DB2 system. This field is produced by ACOVERA and SAOVERA.

PLANNAME
: Plan name of the application program, transaction, or utility executed. This field is produced by ACOVERP and SAOVERP.

# OCCUR
: Number of times the plan was executed or number of plan executions for the user.

ABN. TERM.
: Total number of abnormal terminations per plan execution. This value includes application abends, rollbacks, deadlocks, cancels, and thread abends due to a resource shortage.

AVG ELAPSED TIME
: Average elapsed time in seconds per plan execution.

AVG CPU TIME
: Average CPU time in seconds spent per plan execution. This is the sum of the TCB and SRB times for the application.

AVG SQL MANIP.
Average number of SQL data manipulation statements executed per plan execution. This includes SELECT, INSERT, UPDATE, DELETE, PREPARE, DESCRIBE, OPEN, CLOSE, and FETCH statements.

AVG SQL CONTROL
Average number of SQL CONTROL statements executed per plan execution.

AVG SQL DEFINIT.
Average number of SQL data definition statements executed per plan execution.

AVG GETPAGES
Average number of GETPAGE requests to all buffer pools per plan execution.

AVG SYNC READ I/O
Average number of synchronous read I/Os issued per plan execution.

AVG PREFETCH REQUESTS
Average number of sequential prefetch requests issued per plan execution.

AVG PAGE UPDATES
Average number of pages updated per plan execution.

AVG LOCK SUSPENDS
Average number of times suspended due to waiting for lock per plan execution.

MAX OF MAX PG LOCKS
Maximum number of page locks held in any one plan execution.

# xxOVDFy—DB2 Accounting DDF Overview Report

The DB2 Accounting DDF Overview Report provides an overview of the DDF activity for the DB2 system. It is produced from either detailed accounting data (ACOVDFA and ACOVDFP) or summary accounting data (SAOVDFA and SAOVDFP). It can be plan-oriented (ACOVDFP and SAOVDFP) or user-oriented (ACOVDFA and SAOVDFA).

```
BMC SOFTWARE INC                              DB2 ACCOUNTING DDF OVERVIEW REPORT                                      PAGE 0001
REPORT: ACOVDFA                                                                                REPORT DATE: 2001-03-25 10.25.45
DATE FROM: 2001-03-01  09:13:36                                                                               LOCATION - SANJOSE
DATE TO:   2001-03-19  11:13:17                                                                                SUBSYSTEM - DB2D
                                       AVG      AVG     AVG      AVG      AVG     AVG       AVG      AVG       AVG
                        REMOTE         SQL      SQL     CONV     CONV    TRANS    TRANS    COMMIT   COMMIT     CONV
 AUTHID     #OCCUR    LOCATION        SENT     RECV   INIT FM  INIT TO   MIG FM   MIG TO    SENT     RECV   QUEUED
 --------  -------  --------------  --------  ------- -------- -------- -------- -------- -------- -------- --------
 ACCTDEP         5 1234567890123456    156.4      0.0      0.0    167.6      4.0      0.0      0.0      0.6      0.2
 BABUSER        42 REMLOCNAMEABCDEF     67.1      0.0      1.9    896.3     41.3     51.7    371.5      0.0      3.3
 CD10            5 LOCABC                0.0      0.0      0.0      0.0      0.0      0.0      0.0      0.0      0.0
 CIM5           36                      12.1      0.0      0.0     37.1      5.8      0.8      0.0      0.0      0.7
```

Figure 32.  DB2 Accounting DDF Overview Report

AUTHID
: Authorization ID of the user accessing this DB2 system. This field is produced by ACOVERA and SAOVERA.

PLANNAME
: Plan name of the application program, transaction, or utility executed. This field is produced by ACOVERP and SAOVERP.

#OCCUR
: Number of times the plan was executed or number of plan executions for the user.

REMOTE LOCATION
: Sixteen-character identifier of the remote location.

AVG SQL SENT
: Average number of SQL statements sent.

AVG SQL RECV
: Average number of SQL statements received.

AVG CONV INIT FM
: Average number of conversations initiated from this site.

AVG CONV INIT TO
: Average number of conversations initiated to this site.

AVG TRANS MIG TO
: Average number of transactions migrated to remote locations.

AVG TRANS MIG FM
: Average number of transactions migrated from remote locations.

AVG COMMIT SENT
: Average number of commits sent.

AVG COMMIT RECV
Average number of commits received.

AVG CONV QUEUED
Average number of conversation requests queued.

# ACEXCEP—DB2 Accounting Exceptions Report

The DB2 Accounting Exceptions Report is produced from detailed accounting data. A row is displayed for each thread for which at least one of the listed abnormal conditions was detected. Some of these conditions cause an abnormal termination of the thread.

```
BMC SOFTWARE INC                            DB2 ACCOUNTING EXCEPTIONS REPORT                                        PAGE -     1
REPORT: ACEXCEP                                                                                REPORT DATE: 2001-03-25 17.07.45
DATE FROM: 2001-03-01  10.25.16                                                                              LOCATION - SANJOSE
DATE TO:   2001-03-19  18.46.04                                                                                SUBSYSTEM - DB2F

                                                 ABNORM.                              LOCK     MAX    INCRMNT.   LOCK
DATETIME            AUTHID   PLANNAME  CONNECT  TERMINAT    ABORT DEADLOCK  TIMEOUT SUSPENDS PG LOCKS    BINDS ESCALAT.
------------------- -------- -------- --------  -------- -------- -------- -------- -------- -------- -------- --------
2001-03-01-10.25.16 MXW4              DB2CALL         20        0        0        0        0        0        0        0
2001-03-01-11.16.09 MXW4              DB2CALL         20        0        0        0        0        0        0        0
2001-03-01-11.51.40 MXW2     DSNESPCS TSO             12        1        0        0        0        5        0        0
2001-03-01-13.19.18 MXW4              DB2CALL         20        0        0        0        0        0        0        0
2001-03-01-13.37.41 MXW4              DB2CALL         20        0        0        0        0        0        0        0
2001-03-01-14.31.24 MXW4              DB2CALL         20        0        0        0        0        0        0        0
2001-03-01-16.41.29 HHH3     DSNESPRR TSO             12        1        0        0        0       11        0        0
2001-03-01-16.42.35 MXW4              DB2CALL         20        0        0        0        0        0        0        0
2001-03-01-16.42.53 HHH3     DSNESPRR TSO             12        1        0        0        0        8        0        0
2001-03-01-16.50.07 HHH2              DB2CALL         28        0        0        0        0        0        0        0
2001-03-01-17.06.31 HHH2              DB2CALL         28        0        0        0        0        0        0        0
2001-03-01-18.42.26 MXW4              DB2CALL         20        0        0        0        0        0        0        0
2001-03-01-19.19.26 HHH2              DB2CALL         28        0        0        0        0        0        0        0
2001-03-19-09.32.09 MXW4              DB2CALL         20        0        0        0        0        0        0        0
2001-03-19-11.23.23 CJN1     DSNESPCS TSO             12        1        0        1        1        2        0        0
2001-03-19-11.36.53 CJN2     DSNESPCS TSO             12        1        0        0        0       96        0        0
2001-03-19-12.11.48 MXW4              DB2CALL         20        0        0        0        0        0        0        0
2001-03-19-13.35.27 CJN2     DSNESPCS TSO             12        1        0        0        0        2        0        0
2001-03-19-15.05.31 CJN1     DSNUTIL  UTILITY         20        1        0        0        0        5        0        0
```

Figure 33. DB2 Accounting Exceptions Report

DATETIME
Date and time each thread terminated.

AUTHID
Authorization ID of the user submitting this application.

PLANNAME
Plan name of the application program, transaction, or utility executed.

CONNECT
Connection name.

ABNORM. TERMINAT
If the thread terminated due to abnormal reasons, reflects the abnormal termination reason.

ABORT
Number of aborts for each thread.

DEADLOCK
Number of deadlocks for each thread.

TIMEOUT
Number of timeouts for each thread.

LOCK SUSPENDS
Number of times each thread was suspended because it was waiting for a lock. If this value is greater than 10, it causes the row to be displayed.

**Note:** The threshold value specified here and in the query can be modified to satisfy site requirements.

MAX PG LOCKS
Maximum number of page locks for each thread. If this value is greater than 1000, it causes the row to be displayed.

**Note:** The threshold value specified here and in the query can be modified to satisfy site requirements.

INCRMNT. BINDS
Number of incremental binds for each thread.

LOCK ESCALAT.
Number of lock escalations for each thread.

## User Activity Reports

The user activity reports provide detailed or summary accounting information about SQL processing by user or plan name, dynamic SQL statement cache activity, the amount of CPU consumed and elapsed time waiting for resources within DB2 per plan execution, CPU usage or local or remote activity by DDF, package activity and wait times per package, and parallel I/O activity.

## xxSQLy—DB2 Accounting SQL Report—Averages

The DB2 Accounting SQL Report is produced from either detailed accounting data (ACSQLA and ACSQLP) or summary accounting data (SASQLA and SASQLP). It can be plan-oriented (ACSQLP and SASQLP) or user-oriented (ACSQLA and SASQLA).

```
BMC SOFTWARE INC                              DB2 ACCOUNTING SQL REPORT - AVERAGES                                   PAGE 0002
REPORT: ACSQLA                                                                                REPORT DATE: 2001-03-25 11.14.54
DATE FROM: 2001-03-01 07:29:45                                                                               LOCATION - SANJOSE
DATE TO:   2001-03-19 00:06:28                                                                               SUBSYSTEM - DB2P
                                                                                             OTHER
AUTHID   #OCCUR SELECT INSERT UPDATE DELETE PREPAR  OPEN  FETCH  CLOSE  CREATE  DROP  ALTER   SQL   ROLLBK COMMIT REOPT
-------- ------ ------ ------ ------ ------ ------ ------ ------ ------ ------ ------ ------ ------ ------ ------ ------
BOLCJN2      20    0.0    0.0    0.0    0.0    9.1    8.2 3896.3    7.5    0.0    0.0    0.0    0.0    0.4    0.5    0.0
BOLCJN3      15    0.0    0.0    0.0    0.0    5.0    4.8 3974.4    4.6    0.0    0.0    0.0    0.4    0.4    0.9    0.0
BOLHHH1       4    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.2    0.0    1.0    0.0
BOLJEH1       1    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    1.0    0.0
BOLJEH4       2    0.0    0.0    0.0    0.0    3.5    1.5   77.5    1.5    0.0    0.0    0.0    1.5    0.0    1.5    0.0
BOLJGQ1      63    0.0    0.1    0.2    0.1    0.8    0.6    7.0    0.5    0.3    0.1    0.1    0.0    0.6    3.8    0.0
BOLJGQ2       9    0.0    0.0    1.0    0.0    2.0    1.0    1.0    1.0    0.0    0.0    0.0    0.0    1.4    0.0    0.0
BOLSMR2      13    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.9    0.0
```

Figure 34. DB2 Accounting SQL Report—Averages

AUTHID
: Authorization ID of the user submitting this application. This field is produced by ACSQLA and SASQLA.

PLANNAME
: Plan name of the application program, transaction, or utility executed. This field is produced by ACSQLP and SASQLP.

# OCCUR
: Number of times the plan was executed or number of plan executions for the user.

SELECT
: Average number of SELECT statements processed per plan execution.

INSERT
: Average number of INSERT statements processed per plan execution.

UPDATE
: Average number of UPDATE statements processed per plan execution.

DELETE
: Average number of DELETE statements processed per plan execution.

PREPAR
: Average number of PREPARE statements processed per plan execution.

OPEN
Average number of OPEN statements processed per plan execution.

FETCH
Average number of FETCH statements processed per plan execution.

CLOSE
Average number of CLOSE statements processed per plan execution.

CREATE
Average number of CREATE statements processed per plan execution.

DROP
Average number of DROP statements processed per plan execution.

ALTER
Average number of ALTER statements processed per plan execution.

OTHER SQL
Average number of other SQL statements processed per plan execution.

ROLLBK
Average number of unit of work rollbacks performed per plan execution.

COMMIT
Average number of unit of work commits performed per plan execution.

REOPT
Number of times reoptimization occurred (DB2 5.1 and later only).

## xxDYNCy—DB2 Accounting Dynamic SQL Statement Cache Report

The DB2 Accounting Dynamic SQL Statement Cache Report is produced from either detailed accounting data (ACDYNCA and ACDYNCP) or summary accounting data (SADYNCA and SADYNCP). It can be plan-oriented (ACDYNCP and SADYNCP) or user-oriented (ACDYNCA and SADYNCA). This report is available for DB2 5.1 and later only.

```
 BMC SOFTWARE INC                 DB2 ACCOUNTING DYNC REPORT - AVERAGES                    PAGE -     2
 REPORT: ACDYNCA                                                                REPORT DATE: 2001-03-25 11.31.16
 DATE FROM: 2001-03-01 08.48.18                                                            LOCATION - SANJOSE
 DATE TO:   2001-03-19 21.26.07                                                           SUBSYSTEM - DB2G

                     STMT    STMT   PREPARE IMPLICT STMT     STMT
  AUTHID    #OCCUR  FOUND  NOT FND AVOIDED PREPARE DISCARD PURGED
  -------- ------- ------- ------- ------- ------- ------- -------
  ACCTDEP        5     0.0     0.0     0.0     0.0     0.0     0.0
  BAOSTC         1     0.0     0.0     0.0     0.0     0.0     0.0
  BOLBPL1       90     0.0     0.0     0.0     0.0     0.0     0.0
  BOLBPL2       25     0.0     0.0     0.0     0.0     0.0     0.0
  BOLBPL4        5     0.0     0.0     0.0     0.0     0.0     0.0
  BOLCJN2       20     0.0     0.0     0.0     0.0     0.0     0.0
  BOLCJN3       15     0.0     0.0     0.0     0.0     0.0     0.0
  BOLHHH1        4     0.0     0.0     0.0     0.0     0.0     0.0
```

Figure 35. DB2 Accounting Dynamic SQL Statement Cache Report

AUTHID
: Authorization ID of the user submitting this application. This field is produced by ACDYNCA and SADYNCA.

PLANNAME
: Plan name of the application program, transaction, or utility executed. This field is produced by ACDYNCP and SADYNCP.

# OCCUR
: Number of times the plan was executed or number of plan executions for the user.

STMT FOUND
: Number of times DB2 satisfied a PREPARE request by making a copy of a statement in the prepared statement cache.

STMT NOT FND
: Number of times DB2 searched the prepared statement cache but could not find a suitable prepared statement.

PREPARE AVOIDED
: Number of times DB2 did not PREPARE a statement bound with KEEPDYNAMIC(YES) because the prepared statement cache contained a valid copy of the prepared statement.

IMPLICIT PREPARE
: Number of times DB2 did an implicit PREPARE for a statement bound with KEEPDYNAMIC(YES) because the prepared statement cache did not contain a valid copy of the prepared statement.

STMT DISCARD
: Number of times DB2 discarded a prepared statement from the prepared statement cache because the number of prepared statements in the cache exceeded the value of subsystem parameter MAXKEEPD.

STMT PURGED
: Number of times DB2 discarded a prepared statement from the prepared statement cache because a program executed a DROP, ALTER, or REVOKE statement against a dependent object.

## xxCPUy—DB2 Accounting CPU/Elapsed Time Report

The DB2 Accounting CPU/Elapsed Time Report is produced from either detailed accounting data (ACCPUA and ACCPUP) or summary accounting data (SACPUA and SACPUP). It can be plan-oriented (ACCPUP and SACPUP) or user-oriented (ACCPUA and SACPUA).

```
BMC SOFTWARE INC                                 DB2 ACCOUNTING AVG CPU/ELAPSED TIME REPORT                       PAGE -     4
REPORT: ACCPUA                                                                                REPORT DATE: 2001-03-25 14.48.55
DATE FROM: 2001-03-01   10.25.27                                                                             LOCATION - SANJOSE
DATE TO:   2001-03-19   21.44.17                                                                              SUBSYSTEM - DB2G

                  CLASS 1-->                              CLASS 2-->                              CLASS 3-->
                      AVG      AVG       AVG       AVG       AVG       AVG       AVG     AVG-DB2     AVG       AVG       AVG
                   ELAPSED     TCB       SRB     ROUTINE   ELAPSED     DB2       DB2     ROUTINE     I/O     LK/LATCH   OTHER
AUTHID    #OCCUR     TIME      CPU       CPU       TCB      IN DB2     TCB       SRB       TCB       WAIT      WAIT     WAITS
-------- ------- --------- --------- --------- --------- --------- --------- --------- --------- --------- --------- ---------
BOLBPL1      467      0.31  0.003907  0.000404  0.000000  0.151901  0.001526  0.000226  0.000000  0.000000  0.001481  0.004907
BOLCJN2       31     28.87  3.868025  0.047747  0.000000 23.533142  3.197154  0.014397  0.000000  2.720594  0.007555  9.039436
BOLHHH3        4      0.06  0.023604  0.000477  0.000000  0.063274  0.023167  0.000477  0.000000  0.000000  0.000000  0.000000
BOLMXW3       32      5.76  0.603356  0.036983  0.000000  1.575838  0.158559  0.002177  0.000000  0.169542  0.005260  0.095898
BOLSMR2      116      5.37  0.166765  0.008295  0.005047  4.587118  0.079810  0.000987  0.000045  0.127715  0.000076  1.532541
BOLSMR4      436     11.81  0.540677  0.012574  0.000000 10.336602  0.391044  0.003169  0.000000  0.322712  0.094239  8.254516
```

Figure 36. DB2 Accounting CPU/Elapsed Time Report

AUTHID
Authorization ID of the user submitting this application. This field is produced by ACCPUA and SACPUA.

PLANNAME
Plan name of the application program, transaction, or utility executed. This field is produced by ACCPUP and SACPUP.

# OCCUR
Number of times the plan was executed or number of plan executions for the user.

AVG ELAPSED TIME
Average elapsed time in seconds per plan execution. This value is available with accounting trace level 1.

AVG TCB CPU
Average TCB CPU time consumed in seconds per plan execution. This value is available with accounting trace level 1.

AVG SRB CPU
Average SRB CPU time consumed in seconds per plan execution. This value is available with accounting trace level 1.

AVG ROUTINE TCB
Average TCB CPU time in seconds consumed within DB2 processing SQL statements issued by stored procedures, triggers, and user-defined functions per plan execution. This value is available with accounting trace level 1.

AVG ELAPSED IN DB2
Average elapsed time within DB2 in seconds per plan execution. This value is available with accounting trace level 2.

AVG DB2 TCB
Average TCB CPU time consumed within DB2 in seconds per plan execution. This value is available with accounting trace level 2.

AVG DB2 SRB

Average SRB CPU time consumed within DB2 in seconds per plan execution. This value is available with accounting trace level 2.

AVG-DB2 ROUTINE TCB

Average TCB CPU time in seconds consumed within DB2 processing SQL statements issued by stored procedures, triggers, and user-defined functions per plan execution. This value is available with accounting trace level 2.

AVG I/O WAIT

Average elapsed time waiting for I/O to complete in seconds per plan execution. This value is available with accounting trace level 3.

**Note:** Log waits are not included in this value for DB2 6.1 and later.

AVG LOCK/LATCH WAIT

Average elapsed time waiting to acquire a latch or lock in seconds per plan execution. This value is available with accounting trace level 3.

ENTRY/EXIT EVENTS

Total number of entry or exit events for each user or plan. This value is available with accounting trace level 3.

WAIT EVENTS

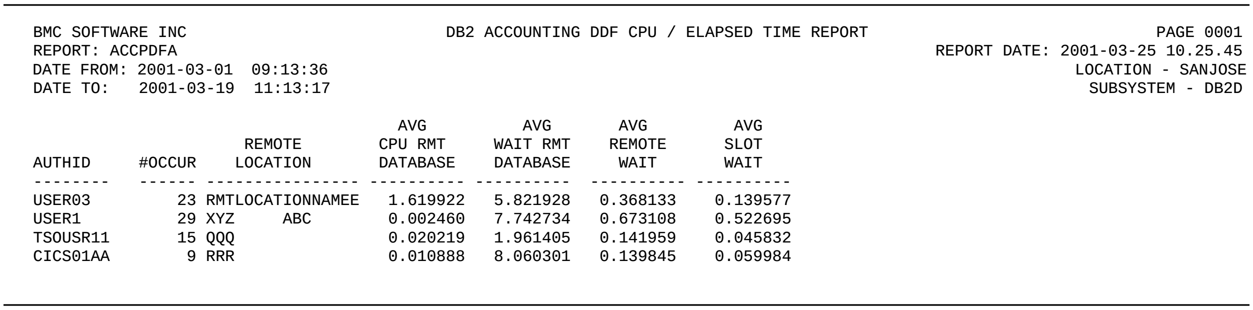
Total number of wait events for each user or plan. This value is available with accounting trace level 3.

AVG OTHER WAITS

Average elapsed time waiting for other class 3 events to complete in seconds per plan execution. This value is available with accounting trace level 3. This includes waits for OTHER READ, OTHER WRITE, UNIT SWITCH FOR DB2 SERVICES, ARCHIVE LOG MODE (QUIESCE) COMMAND, ARCHIVE READ FROM TAPE, DRAIN LOCK, CLAIM RELEASE, and PAGE LATCH CONTENTION; and for DB2 Release 4.1 and later, GLOBAL LOCK CONTENTION, GLOBAL NOTIFY MESSAGES, and STORED PROCEDURE TCB. For DB2 7.1 and later, this also includes waits for GLOBAL CONTENTION FOR CHILD L-LOCKS, OTHER L-LOCKS, PAGE SET OR PARTITION P-LOCKS, PAGE P-LOCKS, and OTHER P-LOCKS.

## xxCPDFy—DB2 Accounting DDF CPU/Elapsed Time Report

The DB2 Accounting DDF CPU/Elapsed Time Report provides an overview of DDF CPU usage. It is produced from either detailed accounting data (ACCPDFA and ACCPDFP) or summary accounting data (SACPDFA and SACPDFP). It can be plan-oriented (ACCPDFP and SACPDFP) or user-oriented (ACCPDFA and SACPDFA).



```
BMC SOFTWARE INC                           DB2 ACCOUNTING DDF CPU / ELAPSED TIME REPORT                              PAGE 0001
REPORT: ACCPDFA                                                                               REPORT DATE: 2001-03-25 10.25.45
DATE FROM: 2001-03-01  09:13:36                                                                             LOCATION - SANJOSE
DATE TO:   2001-03-19  11:13:17                                                                               SUBSYSTEM - DB2D

                                      AVG          AVG       AVG         AVG
                      REMOTE        CPU RMT     WAIT RMT    REMOTE      SLOT
AUTHID     #OCCUR    LOCATION       DATABASE    DATABASE     WAIT       WAIT
--------   ------ ---------------- ---------- ----------  ---------- ----------
USER03         23 RMTLOCATIONNAMEE   1.619922   5.821928   0.368133    0.139577
USER1          29 XYZ     ABC        0.002460   7.742734   0.673108    0.522695
TSOUSR11       15 QQQ                0.020219   1.961405   0.141959    0.045832
CICS01AA        9 RRR                0.010888   8.060301   0.139845    0.059984
```

Figure 37. DB2 Accounting DDF CPU/Elapsed Time Report

AUTHID
: Authorization ID of the user accessing this DB2 system. This field is produced by ACOVERA and SAOVERA.

PLANNAME
: Plan name of the application program, transaction, or utility executed. This field is produced by ACOVERP and SAOVERP.

#OCCUR
: Number of times the plan was executed or number of plan executions for the user.

REMOTE LOCATION
: Sixteen-character location name of the remote site.

AVG CPU RMT DATABASE
: CPU time in remote database.

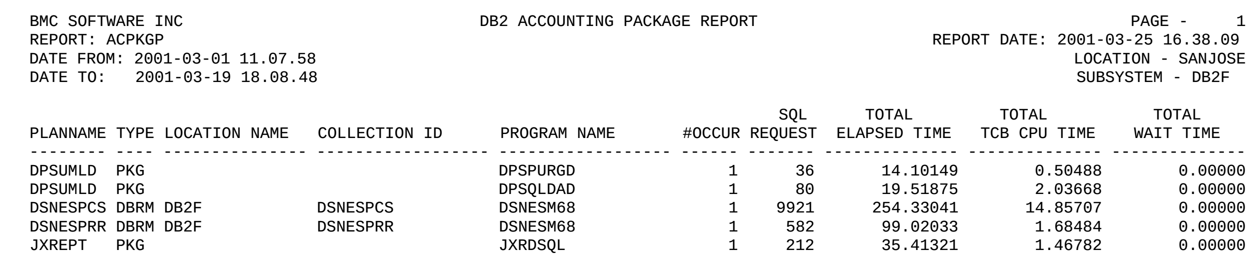
AVG WAIT RMT DATABASE
: Elapsed wait for remote database.

AVG REMOTE WAIT
: Elapsed time remote wait.

AVG SLOT WAIT
: Total elapsed time spent waiting for an available database access agent slot (DB2 5.1 and later only). This wait occurs when DB2 reaches its maximum number of database access agents, and a DBAT must wait for another DBAT to relinquish its slot.

# xxPKGy—DB2 Accounting Package Report

The DB2 Accounting Package Report provides an overview of package activity. It is produced from either detailed accounting package data (ACPKGA and ACPKGP) or summary accounting package data (SAPKGA and SAPKGP). It can be plan-oriented (ACPKGP and SAPKGP) or user-oriented (ACPKGA and SAPKGA).



```
BMC SOFTWARE INC                              DB2 ACCOUNTING PACKAGE REPORT                                 PAGE -     1
REPORT: ACPKGP                                                                                 REPORT DATE: 2001-03-25 16.38.09
DATE FROM: 2001-03-01 11.07.58                                                                                LOCATION - SANJOSE
DATE TO:   2001-03-19 18.08.48                                                                                SUBSYSTEM - DB2F

                                                                                SQL      TOTAL         TOTAL          TOTAL
PLANNAME TYPE LOCATION NAME   COLLECTION ID      PROGRAM NAME       #OCCUR REQUEST  ELAPSED TIME   TCB CPU TIME    WAIT TIME
-------- ---- -------------- ------------------ ------------------ ------ ------- -------------- -------------- --------------
DPSUMLD  PKG                                    DPSPURGD                1      36       14.10149        0.50488        0.00000
DPSUMLD  PKG                                    DPSQLDAD                1      80       19.51875        2.03668        0.00000
DSNESPCS DBRM DB2F           DSNESPCS           DSNESM68                1    9921      254.33041       14.85707        0.00000
DSNESPRR DBRM DB2F           DSNESPRR           DSNESM68                1     582       99.02033        1.68484        0.00000
JXREPT   PKG                                    JXRDSQL                 1     212       35.41321        1.46782        0.00000
```

Figure 38. DB2 Accounting Package Report

AUTHID
: Authorization ID of the user accessing this DB2 system. This field is produced by ACPKGA and SAPKGA.

PLANNAME
: Plan name of the application program, transaction, or utility executed. This field is produced by ACPKGP and SAPKGP.

TYPE
: DBRM, PKG, or BOTH. Indicates a package or DBRM. BOTH can appear if there are packages and DBRMs with the same program name.

LOCATION NAME
: Location name. If blank, the package or DBRM executed locally. If it is not blank, all times represent the time spent locally to execute the remote package for this requester.

COLLECTION ID
: Package collection ID. For DBRMs, this field is not applicable.

PROGRAM NAME
: Program name (package ID or DBRM).

#OCCUR
: Number of accounting records with accounting data for this package.

SQL REQUEST
: Number of SQL statements issued in this package or DBRM. This field might not equal the total of all SQL statement counters in the QXST data section because QXST does not count all SQL statements. For example, it does not count commit or rollback statements.

TOTAL ELAPSED TIME
: Total elapsed time spent processing all executions of this package or DBRM.

TOTAL TCB CPU TIME
: Total TCB CPU time for all executions of this package or DBRM.

TOTAL WAIT TIME
: Total time the program spent waiting for all types of suspensions.

# xxPKGS—DB2 Accounting Package Suspensions Report

The DB2 Accounting Package Suspensions Report shows wait times per package. It is produced from either detailed accounting suspensions data (ACPKGS) or summary accounting suspensions data (SAPKGS).

```
BMC SOFTWARE INC                            DB2 ACCOUNTING PACKAGE SUSPENSIONS REPORT                              PAGE -     3
REPORT: ACPKGS                                                                                REPORT DATE: 2001-03-25 14.48.55
DATE FROM: 2001-03-01 10.25.27                                                                               LOCATION - SANJOSE
DATE TO:   2001-03-19 16.05.42                                                                               SUBSYSTEM - DB2G

                 <    TOTAL   > < LOCK/LATCH > <   SYNC-IO   > < OTHER-READS > <   ROUTINES  > < DATA SHARING> <   OTHER     >
PROGRAM NAME     EVENT  AVGTIME  EVENT AVGTIME  EVENT  AVGTIME  EVENT  AVGTIME  EVENT  AVGTIME  EVENT  AVGTIME  EVENT  AVGTIME
--------------- ------- ------- ------ ------- ------- ------- ------- ------- ------- ------- ------- ------- ------- -------
DSNCAL1              18    0.04      0    0.00      12    0.00       0    0.00       0    0.00       0    0.00       6    0.13
DSNCAL2              34    0.08      0    0.00      10    0.01       0    0.00       0    0.00       0    0.00      24    0.11
DSNESM68           1212    0.13     50    0.00     770    0.00      66    0.02       0    0.00     102    0.52     224    0.43
DSNTEP2            6362    0.10    312    0.02    3002    0.01    1640    0.03       0    0.00     822    0.21     586    0.63
DSN8CC0              24    0.00      4    0.00      20    0.01       0    0.00       0    0.00       0    0.00       0    0.00
DSN8CC1             896    0.10     22    3.14     448    0.01       0    0.00       0    0.00       0    0.00     426    0.04
DSN8CC2              94    0.01      4    0.00      72    0.01       0    0.00       0    0.00       0    0.00      18    0.03
DSN8EP1              12    0.01      0    0.00       0    0.00       0    0.00       0    0.00       0    0.00      12    0.01
RXSEL1M            1688    0.08    126    0.00     658    0.01     602    0.02       0    0.00      16    3.81     286    0.18
```

Figure 39. DB2 Accounting Package Suspensions Report

PROGRAM NAME
Program name (package ID or DBRM).

TOTAL EVENT
Total number of events.

TOTAL AVG TIME
Total average time for all events.

LOCK/LATCH EVENT
Total number of lock and latch wait trace events while executing this package or DBRM. This value includes

- Lock and latch suspensions
- Drain locks
- Claims
- Page latch contention

LOCK/LATCH AVG TIME
Average elapsed wait time for lock and latch suspensions, drain locks, drains when waiting for claims to be released, and page latch contention while executing this page or DBRM.

SYNC-IO EVENT
Number of wait trace events processed for waits for I/O under this thread while executing this package or DBRM.

SYNC-IO AVG TIME
Average I/O elapsed wait time per wait trace events for I/O done under this thread while executing this package or DBRM.

OTHER-READS EVENT
Number of wait trace events processed for waits for read I/O under another thread while executing this package or DBRM.

OTHER-READS AVG TIME

Average wait time for read I/O done under a thread other than this one while executing this package or DBRM.

ROUTINES EVENT

Number of wait trace events processed for waits for an available TCB before a stored procedure or user-defined function could be scheduled while executing this package or DBRM.

ROUTINES AVG TIME

Average wait time for an available TCB before a stored procedure or user-defined function could be scheduled while executing this package or DBRM.

DATA SHARING EVENT

Number of wait trace events processed for waits for data sharing (global lock contention or notify messages) while executing this package or DBRM.

DATA SHARING AVG TIME

Average wait time for data sharing (global lock contention or notify messages) while executing this package or DBRM.

OTHER EVENT

Number of wait trace events processed for waits for other reasons while executing this package or DBRM. This includes waits for

- Write I/O under another thread
- Processing archive log mode (quiesce) commands
- Processing archive log reads
- Synchronous execution unit switch to DB2 services

OTHER AVG TIME

Average wait time for other reasons while executing this package or DBRM. This includes waits for

- Write I/O under another thread
- Processing archive log mode (quiesce) commands
- Processing archive log reads
- Synchronous execution unit switch to DB2 services

  The following DB2 service waits are included in this field:

  - Open and close of a data set
  - Update of SYSLGRNG
  - HSM recall data set
  - Dataspace manager services

# xxPLIOy—DB2 Accounting Parallelism Report

The DB2 Accounting Parallelism Report provides an overview of parallel query activity, including both I/O and CPU parallelism. It is produced from either detailed accounting data (ACPLIOA and ACPLIOP) or summary accounting data (SAPLIOA and SAPLIOP). It can be plan-oriented (ACPLIOP and SAPLIOP) or user-oriented (ACPLIOA and SAPLIOA).

```
BMC SOFTWARE INC                            DB2 ACCOUNTING PARALLELISM REPORT                                       PAGE -     4
REPORT: ACPLIOA                                                                                       REPORT DATE: 2001-03-25 14.48.55
DATE FROM: 2001-03-01  10.25.27                                                                                  LOCATION - SANJOSE
DATE TO:   2001-03-19  21.44.17                                                                                   SUBSYSTEM - DB2G

                             <GRPS PROCESS> <-------GROUP FALLBACK REASON---->              <------SYSPLEX PARALLELISM------->
                                                                                                                          OTHER
                   MAX  GROUPS PLAN   REDUCE   NO   UPDATE NO ESA   NO    RLF   SET CR GETPG  1 DB2  1 DB2   DB2   GROUPS  PRLL
AUTHID   #OCCUR DEGREE EXEC   DEGREE DEGREE BUFFER CURSOR  SORT  ENCLV  DISABL DEGREE FAILED COR=NO ISOL.  BYPASS PLAND  CHANGE
-------- ------ ------ ------ ------ ------ ------ ------ ------ ------ ------ ------ ------ ------ ------ ------ ------ ------
BMCDWW1      76      2      5      2      3      3      0      0      0      0     10      0      0      0      3      5      0
```

Figure 40. DB2 Accounting Parallelism Report

AUTHID
: Authorization ID of the user accessing this DB2 system. This field is produced by ACPLIOA and SAPLIOA.

PLANNAME
: Plan name of the application program, transaction, or utility executed. This field is produced by ACPLIOP and SAPLIOP.

#OCCUR
: Number of times the plan was executed or number of plan executions for the user.

MAX DEGREE
: Maximum degree of parallelism among the parallel groups to indicate the extent to which query parallelism applies.

GROUPS EXEC
: Total number of parallel groups executed.

GRPS PROCESS PLAN DEGREE
: Total number of parallel groups with a planned degree greater than one at run time, and were executed to the same degree because of sufficient storage on the buffer pool.

GRPS PROCESS REDUCE DEGREE
: Total number of parallel groups with a planned degree greater than one at run time, but were processed to a parallel degree less than planned because of a storage shortage or contention on the buffer pool.

GROUP FALLBACK REASON NO BUFFER
: Total number of parallel groups with a planned degree greater than one at run time, but fell back to sequential mode because of storage shortage or contention on the buffer pool.

GROUP FALLBACK REASON UPDATE CURSOR
: Total number of parallel groups that fell back to sequential mode because the cursor may be used in UPDATE/DELETE.

GROUP FALLBACK REASON NO ESA SORT

Total number of parallel groups that fell back to sequential mode due to lack of ESA sort support.

GROUP FALLBACK REASON NO ENCLV

Total number of parallel groups that fell back to sequential mode due to lack of MVS/ESA 5.2 SRB enclave services.

GROUP FALLBACK REASON RLF DISABL

If non-zero, indicates query parallelism was disabled by the resource limit facility for at least one dynamic select statement within a thread.

SET CR DEGREE

Number of SET CURRENT DEGREE statements executed.

GETPG FAILED

Number of times conditional GETPAGE requests could not be satisfied for this buffer pool.

SYSPLEX PARALLELISM 1 DB2 COR=NO

Number of parallel groups executed on a single DB2 because the coordinator value is set to NO (DB2 5.1 and later only).

SYSPLEX PARALLELISM 1 DB2 ISOL.

Total number of parallel groups executed on a single DB2 because the plan or package was bound with an isolation value of repeatable read (DB2 5.1 and later only).

SYSPLEX PARALLELISM DB2 BYPASS

Number of times the parallelism coordinator had to bypass a DB2 because of not enough buffer pool storage (DB2 5.1 and later only).

SYSPLEX PARALLELISM GROUPS PLAND

Number of parallel groups DB2 intended to run across the data sharing group (DB2 5.1 and later only).

SYSPLEX PARALLELISM OTHER PRLL CHANGE

Number of parallel groups for which DB2 reformulated the parallel portion of the access path (DB2 6.1 and later only) for one of these reasons:

– The SYSPLEX configuration at run time was different from the SYSPLEX configuration at bind time.

– There was not enough buffer pool resource.

– A query block used a user-defined function with a Declared Temporary Table (DB2 7.1 and later only).

## xxSPRCy—DB2 Accounting Stored Procedures Report

The DB2 Accounting Stored Procedures Report provides information about stored procedure activity. It is produced from either detailed accounting data (ACSPRCA and ACSPRCP) or summary accounting data (SASPRCA and SASPRCP). It can be plan-oriented (ACSPRCP and SASPRCP) or user-oriented (ACSPRCA and SASPRCA).

```
BMC SOFTWARE INC                        DB2 ACCOUNTING STORED PROCEDURES REPORT                                    PAGE -     4
REPORT: ACSPRCA                                                                                        REPORT DATE: 2001-03-25 14.48.55
DATE FROM: 2001-03-01 10.25.27                                                                                       LOCATION - SANJOSE
DATE TO:   2001-03-19 21.44.17                                                                                       SUBSYSTEM - DB2G

                  <----------TOTALS-------------------------> <----------AVERAGES------------------------>
AUTHID    #OCCUR   CALLS    ABENDS  TIMEOUTS REJECTS  TCBWAITS  CALLS    ABENDS  TIMEOUTS  REJECTS TCBWAITS
-------- -------- -------- -------- -------- -------- -------- -------- -------- -------- -------- --------
BOLCJN2       31        0        0        0        0        0     0.00    0.000    0.000    0.000    0.000
BOLCJN3        4        0        0        0        0        0     0.00    0.000    0.000    0.000    0.000
BOLHHH3        4        0        0        0        0        0     0.00    0.000    0.000    0.000    0.000
BOLHHH4        8        0        0        0        0        0     0.00    0.000    0.000    0.000    0.000
BOLMXW1        1        0        0        0        0        0     0.00    0.000    0.000    0.000    0.000
BOLMXW3       32        0        0        0        0        0     0.00    0.000    0.000    0.000    0.000
BOLSMR2      116       20        7        1        0        0     0.17    0.060    0.008    0.000    0.000
BOLSMR3       30        0        0        0        0        0     0.00    0.000    0.000    0.000    0.000
BOLSMR4      436       28        0        0        0        0     0.06    0.000    0.000    0.000    0.000
```

Figure 41. DB2 Accounting Stored Procedures Report

AUTHID
Authorization ID of the user accessing this DB2 system. This field is produced by ACSPRCA and SASPRCA.

PLANNAME
Plan name of the application program, transaction, or utility executed. This field is produced by ACSPRCP and SASPRCP.

#OCCUR
Number of times the plan was executed or number of plan executions for the user.

TOTAL CALLS
Total number of stored procedure CALL statements executed.

TOTAL ABENDS
Total number of times a stored procedure terminated abnormally.

TOTAL TIMEOUTS
Total number of times an SQL CALL statement timed out while waiting to be scheduled.

TOTAL REJECTS
Total number of times an SQL CALL statement was rejected because the procedure was in the ‘STOP ACTION(REJECT)’ state.

TOTAL TCBWAITS
Total number of times an SQL CALL statement had to wait for an available TCB before the stored procedure could be scheduled. This value is available with accounting class 3.

AVERAGE CALLS
Average number of stored procedure CALL statements executed.

AVERAGE ABENDS
Average number of times a stored procedure terminated abnormally.

AVERAGE TIMEOUTS

Average number of times an SQL CALL statement timed out while waiting to be scheduled.

AVERAGE REJECTS

Average number of times an SQL CALL statement was rejected because the procedure was in the ‘STOP ACTION(REJECT)’ state.

AVERAGE TCBWAITS

Average number of times an SQL CALL statement had to wait for an available TCB before the stored procedure could be scheduled. This value is available with accounting class 3.

## xxDDFLy—DB2 Accounting DDF Local Report

The DB2 Accounting DDF Local Report provides an overview of DDF local activity. It is produced from either detailed accounting data (ACDDFLA and ACDDFLP) or summary accounting data (SADDFLA and SADDFLP). It can be plan-oriented (ACDDFLP and SADDFLP) or user-oriented (ACDDFLA and SADDFLA).

```
BMC SOFTWARE INC                      DB2 ACCOUNTING DDF LOCAL REPORT                                             PAGE 0001
REPORT: ACDDFLA                                                                               REPORT DATE: 2001-03-25 10.25.45
DATE FROM: 2001-03-01  09:13:36                                                                               LOCATION - SANJOSE
DATE TO:   2001-03-19  11:13:17                                                                                 SUBSYSTEM - DB2D
                                    AVG      AVG      AVG      AVG      AVG      AVG      AVG      AVG      AVG      AVG
                     REMOTE         SQL      ROWS     BYTES    CONV     MSGS     TRANS    COMMIT   ABORT    BLKS     CONV
AUTHID     #OCCUR    LOCATION       RECV     RECV     RECV    INIT FM   RECV     MIG FM   RECV     RECV     RECV    QUEUED
-------- -------- ---------------- -------- -------- -------- -------- -------- -------- -------- -------- -------- --------
ACCTDEP         5 1234567890123456    156.4      0.0      0.0    167.6      4.0      0.0      0.0      0.6      0.2      0.2
BABUSER        42 REMLOCNAMEABCDEF     67.1      0.0      1.9    896.3     41.3     51.7    371.5      0.0      3.3      3.3
CD10            5 LOCABC                0.0      0.0      0.0      0.0      0.0      0.0      0.0      0.0      0.0      0.0
CIM5           36                      12.1      0.0      0.0     37.1      5.8      0.8      0.0      0.0      0.7      0.7
```

Figure 42. DB2 Accounting DDF Local Report

AUTHID
: Authorization ID of the user accessing this DB2 system. This field is produced by ACOVERA and SAOVERA.

PLANNAME
: Plan name of the application program, transaction, or utility executed. This field is produced by ACOVERP and SAOVERP.

# OCCUR
: Number of times the plan was executed or number of plan executions for the user.

REMOTE LOCATION
: Sixteen-character location name of the remote site.

AVG SQL RECV
: Average number of SQL statements received.

AVG ROWS RECV
: Average number of rows received.

AVG BYTES RECV
: Average number of bytes received.

AVG CONV INIT FM
: Average number of conversations initiated from this site.

AVG MSGS RECV
: Average number of messages received.

AVG TRANS MIG FM
: Average number of transactions migrated from remote locations.

AVG COMMIT RECV
: Average number of commits received.

AVG ABORT RECV
Average number of abort requests received.

AVG BLKS RECV
Average number of blocks received using block fetch.

AVG CONV QUEUED
Average number of conversation requests queued.

## xxDDFRy—DB2 Accounting DDF Remote Report

The DB2 Accounting DDF Remote Report provides an overview of the DDF remote activity. It is produced from either detailed accounting data (ACDDFRA and ACDDFRP) or summary accounting data (SADDFRA and SADDFRP). It can be plan-oriented (ACDDFRP and SADDFRP) or user-oriented (ACDDFRA and SADDFRA).

```
BMC SOFTWARE INC                           DB2 ACCOUNTING DDF REMOTE REPORT                                         PAGE 0001
REPORT: ACDDFRA                                                                                      REPORT DATE: 2001-03-25 10.25.45
DATE FROM: 2001-03-01 09:13:36                                                                                    LOCATION - SANJOSE
DATE TO:   2001-03-19 11:13:17                                                                                    SUBSYSTEM - DB2D
                                      AVG      AVG      AVG      AVG      AVG      AVG       AVG      AVG      AVG     AVG
                         REMOTE       SQL      ROWS    BYTES     CONV     MSGS     TRANS    COMMIT    ABORT     BLKS    CONV
AUTHID     #OCCUR    LOCATION        SENT     SENT     SENT   INIT TO    SENT     MIG TO    SENT     SENT     SENT   QUEUED
-------- -------- ---------------- -------- -------- -------- -------- -------- -------- -------- -------- -------- --------
ACCTDEP         5 1234567890123456    156.4      0.0      0.0    167.6      4.0      0.0      0.0      0.6      0.2      0.2
BABUSER        42 REMLOCNAMEABCDEF     67.1      0.0      1.9    896.3     41.3     51.7    371.5      0.0      3.3      3.3
CD10            5 LOCABC                0.0      0.0      0.0      0.0      0.0      0.0      0.0      0.0      0.0      0.0
CIM5           36                      12.1      0.0      0.0     37.1      5.8      0.8      0.0      0.0      0.7      0.7
```

Figure 43. DB2 Accounting DDF Remote Report

AUTHID
Authorization ID of the user accessing this DB2 system. This field is produced by ACOVERA and SAOVERA.

PLANNAME
Plan name of the application program, transaction, or utility executed. This field is produced by ACOVERP and SAOVERP.

# OCCUR
Number of times the plan was executed or number of plan executions for the user.

REMOTE LOCATION
Sixteen-character location name of the remote site.

AVG SQL SENT
Average number of SQL statements sent.

AVG ROWS SENT
Average number of rows sent.

AVG BYTES SENT
Average number of bytes sent.

AVG CONV INIT TO
Average number of conversations initiated to this site.

AVG MSGS SENT
Average number of messages sent.

AVG TRANS MIG TO
Average number of transactions migrated to remote locations.

AVG COMMIT SENT
Average number of commits sent.

AVG ABORT SENT
Average number of abort requests sent.

AVG BLKS SENT
Average number of blocks sent using block fetch.

AVG CONV QUEUED
Average number of conversation requests queued.

## Lock Reports

The lock reports provide detailed or summary accounting information about lock activity and suspensions per plan execution.

## xxLOCKy—DB2 Accounting Lock/Latch Report

The DB2 Accounting Lock/Latch Report is produced from either detailed accounting data (ACLOCKA and ACLOCKP) or summary accounting data (SALOCKA and SALOCKP). It can be plan-oriented (ACLOCKP and SALOCKP) or user-oriented (ACLOCKA and SALOCKA). It includes both averages and totals.

```
BMC SOFTWARE INC                                  DB2 ACCOUNTING LOCK/LATCH REPORT                                  PAGE -     1
REPORT: ACLOCKA                                                                                 REPORT DATE: 2001-03-25 14.34.33
DATE FROM: 2001-03-01  00.22.37                                                                               LOCATION - SANJOSE
DATE TO:   2001-03-19  12.28.32                                                                                 SUBSYSTEM - DB2E

                  MAX      AVG     AVG     AVG     AVG
                  OF MAX   LOCK    LOCK   LATCH    OTHER                  ESCALA.ESCALA. LOCKTBL DRAIN-REQUESTS  CLAIM-REQUESTS
AUTHID    #OCCUR PG LCKS REQUSTS SUSPNDS SUSPNDS SUSPNDS DEADLOK TIMEOUT  SHARED EXCLUS.  STMTS  TOTAL   FAIL    TOTAL   FAIL
-------- ------- -------  ------- ------- ------- ------- ------- ------- ------- ------- ------- ------- ------- ------- -------
ACCTDEP       16       2     2.6     0.0     0.0     0.0       0       0       0       0       0       0       0       0       0
CJN1          32      43   459.3     0.0    42.9     0.0       0       0       0       0       0       0       0       0       0
ELC2           1      13  5040.0     0.0     6.0     0.0       0       0       0       2       0       0       0       3       1
NHJ1           7      15    25.2     0.0     1.1     0.0       0       0       0       0       0       0       0       0       0
OLTC           2       0     0.0     0.0     0.0     0.0       0       0       0       0       0       0       0       0       0
TSM5           2       1     3.5     0.0     0.0     0.0       0       0       0       0       0       0       0       0       0
CONTSTC     1548      10    40.7     0.0     1.0     0.0       0       0       0       0       0       0       0       0       0
CPS40          1       6    14.0     0.0     1.0     0.0       0       0       0       0       0       0       0       0       0
```

Figure 44. DB2 Accounting Lock/Latch Report

AUTHID
Authorization ID of the user submitting this application. This field is produced by ACLOCKA and SALOCKA.

PLANNAME
Plan name of the application program, transaction, or utility executed. This field is produced by ACLOCKP and SALOCKP.

# OCCUR
Number of times the plan was executed or number of plan executions for the user.

MAX OF MAX PG LOCKS
Maximum number of page locks held in any one plan execution.

AVG LOCK REQUSTS
Average number of lock requests per plan execution.

AVG LOCK SUSPNDS
Average number of times suspended due to waiting for lock per plan execution.

AVG LATCH SUSPNDS
Average number of times suspended due to latching per plan execution.

AVG OTHER SUSPNDS
Average number of times suspended due to other reasons per plan execution.

DEADLOK
Total number of deadlocks detected.

TIMEOUT
Total number of lock timeouts.

ESCALA. SHARED
Total number of lock escalations in shared mode.

ESCALA. EXCLUS.
Total number of lock escalations in exclusive mode.

LOCKTBL STMTS
Total number of SQL LOCK TABLE statements issued.

DRAIN-REQUESTS TOTAL
Total number of drain requests.

DRAIN-REQUESTS FAIL
Total number of unsuccessful drain requests.

CLAIM-REQUESTS TOTAL
Total number of claim requests.

CLAIM-REQUESTS FAIL
Total number of unsuccessful claim requests.

# xxGLOKy—DB2 Accounting Global Lock Report

The DB2 Accounting Global Lock Report provides information about global lock activity in a data sharing environment. It is produced from either detailed accounting data (ACGLOKA and ACGLOKP) or summary accounting data (SAGLOKA and SAGLOKP). It can be plan-oriented (ACGLOKP and SAGLOKP) or user-oriented (ACGLOKA and SAGLOKA).

```
BMC SOFTWARE INC                            DB2 ACCOUNTING GLOBAL LOCK REPORT                                     PAGE -     4
REPORT: ACGLOKA                                                                                  REPORT DATE: 2001-03-25 14.48.55
DATE FROM: 2001-03-01 10.25.27                                                                                 LOCATION - SANJOSE
DATE TO:   2001-03-19 21.44.17                                                                                 SUBSYSTEM - DB2G

                  <--------PLOCKS----------> <---------XES------------> <-------SUSPENDS---------> INCOMP   NOTIFY
AUTHID     #OCCUR   LOCK    UNLOCK   CHANGE    LOCK    UNLOCK   CHANGE    IRLM      XES     FALSE   RETAINED MSG SENT
-------- -------- -------- -------- -------- -------- -------- -------- -------- -------- -------- -------- --------
BOLBPL1       467        3        0        1       18        4        1        2        3        3        0        0
BOLBPL4        20        0        0        0        0        0        0        0        0        0        0        0
BOLCJN2        31      113       11        8      525      231       14       12        8       18        0       13
BOLCJN3         4        0        0        0       16        4        0        0        0        0        0        0
BOLHHH2        13        0        0        0        0        0        0        0        0        0        0        0
BOLHHH3         4        0        0        0        0        0        0        0        0        0        0        0
BOLHHH4         8        8        0        0       80       22        0        0        0        0        0        0
BOLMXW1         1        0        0        0        0        0        0        0        0        0        0        0
BOLMXW3        32      227      168       22      463      265       23       16       25       21        0        3
BOLSMR2       116      582      146       89     1579      779       92      209       87       80        0        0
BOLSMR3        30        0        0        0        0        0        0        0        0        0        0        0
BOLSMR4       436     4275     1843      231    26925    16537      495      472      132      585        0     3444
DSN8410         9       22        0        0     1975      269      211        0        0        0        0        0
```

Figure 45. DB2 Accounting Global Lock Report

A physical lock (P-lock) is a lock used only by data sharing and is acquired by DB2 to provide consistency on data cached in different DB2 subsystems. P-locks are owned by the subsystem, not by the transaction. XES is the cross-system extended services component of MVS.

AUTHID
: Authorization ID of the user accessing this DB2 system. This field is produced by ACGLOKA and SAGLOKA.

PLANNAME
: Plan name of the application program, transaction, or utility executed. This field is produced by ACGLOKP and SAGLOKP.

#OCCUR
: Number of times the plan was executed or number of plan executions for the user.

PLOCKS-LOCK
: Number of lock requests for P-locks.

PLOCKS-UNLOCK
: Number of unlock requests for P-locks.

PLOCKS-CHANGE
: Number of change requests for P-locks.

XES-LOCK
: Total number of lock requests propagated to MVS XES synchronously under the user’s execution unit. This includes both L-Locks and P-locks. It is not incremented if the request is suspended before going to XES.

XES-UNLOCK

Total number of change requests propagated to MVS XES synchronously under the user's execution unit. This includes both L-Locks and P-locks. It is not incremented if the request is suspended before going to XES.

XES-CHANGE

Total number of resources propagated to MVS XES synchronously under the user's execution unit for unlock requests. This includes both L-Locks and P-locks. It is not incremented if the request is suspended before going to XES. It can be incremented each time a global lock is released by the generic unlock.

SUSPENDS-IRLM

Total number of suspends caused by IRLM global resource contention (IRLM lock states were in conflict on the same resource.). Global contention requires inter-system communication to resolve the lock conflict. This value plus the two other suspends gives the total number of suspends caused by global contention.

SUSPENDS-XES

Total number of suspends caused by MVS XES global resource contention (MVS XES lock states were in conflict but IRLM lock states were not). IRLM has many lock states, but XES is aware only of the exclusive and shared lock states.

SUSPENDS-FALSE

Total number of suspends because of false contentions. This occurs when different resource names hash to the same entry in the coupling facility lock table. This causes MVS XES to detect contention on the hash class. When MVS XES determines that there is no real conflict on the resource, the contention is called *false*.

INCOMP RETAINED

Total number of global lock or change requests denied because of an incompatible retained lock.

NOTIFY MESSAGE SENT

Total number of notify messages sent.

## Buffer Pool Reports

The buffer pool reports provide detailed or summary accounting information about the total or average activity of all buffer pools or a single buffer pool and RID (record ID) pool usage.

## xxBFxTy—DB2 Accounting Buffer Pool Report—Totals

The DB2 Accounting Buffer Pool Report provides information about the total activity of all buffer pools (ACBFATA, ACBFATP, SABFATA, and SABFATP) or a single buffer pool (ACBFXTA, ACBFXTP, SABFXTA, and SABFXATP).

It is produced from either detailed accounting data (ACBFATA, ACBFATP, ACBFXTA, and ACBFXTP) or summary accounting data (SABFATA, SABFATP, SABFXTA, and SABFXTP). It can be plan-oriented (ACBFATP, SABFATP, ACBFXTP, and SABFXTP) or user-oriented (ACBFATA, SABFATA, ACBFXTA, and SABFXTA).

```
BMC SOFTWARE INC                     DB2 ACCOUNTING BUFFER POOL(ALL) REPORT - TOTALS                               PAGE -     2
REPORT: ACBFATA                                                                               REPORT DATE: 2001-03-25 15.59.32
DATE FROM: 2001-03-01 10.25.16                                                                                LOCATION - SANJOSE
DATE TO:   2001-03-19 18.46.04                                                                               SUBSYSTEM - DB2F

                TOTAL     SYNC    PAGES     SYNC    < PREFETCH REQUESTS >  ASYNC   GETPG  < HP READS >  HP->VP  < HP WRITES >
AUTHID  #OCCUR  GETPAGE   READ IO UPDATE    WRT IO  SEQ.   LIST   DYNAMIC PG READ FAILED  TOTAL  FAILED PAGES   TOTAL  FAILED
------- ------- --------- ------- -------- -------- ----- ------- ------- -------- ------ ------ ------ ------- ------ ------
CIR2X         2         0       0        0        0     0       0       0        0      0      0      0       0      0      0
CJN1         41      9815     956     1199       29    32       0      30      268      0      0      0       0      0      0
CJN2         21      4267     433      528        0   132       0       0      185      0      0      0       0      0      0
HHH2          4         0       0        0        0     0       0       0        0      0      0      0       0      0      0
HHH3         15       830      54       85        1    24       0       0       64      0      0      0       0      0      0
MXW2         14     18253     370     1272      455     0       0       0        0      0      0      0     168      0      0
MXW4         14         0       0        0        0     0       0       0        0      0      0      0       0      0      0
NHJ1          5      4342     147       30        0   318       0       0     1680      0      0      0       0      0      0
NHJ1X         7      9004     352        0        0   913       0      14     4965      0      0      0       0      0      0
```

Figure 46. DB2 Accounting Buffer Pool Report—Totals

AUTHID
: Authorization ID of the user accessing this DB2 system. This field is produced by ACBFpTA and SABFpTA.

PLANNAME
: Plan name of the application program, transaction, or utility executed. This field is produced by ACBFpTP and SABFpTP.

#OCCUR
: Number of times the plan was executed or number of plan executions for the user.

TOTAL GETPAGE
: Total number of GETPAGE requests issued. This includes conditional and non-conditional, successful and unsuccessful requests.

SYNC READ IO
: Number of synchronous read I/Os.

PAGES UPDATE
: Number of times a buffer update occurs for the agent.

SYNC WRT IO
: Number of immediate (synchronous) write I/Os.

PREFETCH REQUESTS SEQ.
Number of sequential prefetch requests.

PREFETCH REQUESTS LIST
Number of list prefetch requests.

PREFETCH REQUESTS DYNAMIC
Number of dynamic prefetch requests.

ASYNC PG READ
Number of asynchronous pages read by prefetch under the control of the agent.

GETPAGE FAILED
Number of times conditional GETPAGE requests could not be satisfied for this buffer pool.

HP READS TOTAL
Total number of successful hiperpool reads.

HP READS FAILED
Number of unsuccessful hiperpool reads.

HP->VP PAGES
Number of pages that were found in a hiperpool and moved to a virtual buffer pool because of a prefetch under the control of the agent.

HP WRITES TOTAL
Total number of successful hiperpool writes.

HP WRITES FAILED
Number of unsuccessful hiperpool writes.

## xxBFxAy—DB2 Accounting Buffer Pool Report—Averages

The DB2 Accounting Buffer Pool Report provides information about the average activity of all buffer pools (ACBFAAA, ACBFAAP, SABFAAA, and SABFAAP) or a single buffer pool (ACBFXAA, ACBFXAP, SABFXAA, and SABFXAAP).

It is produced from either detailed accounting data (ACBFAAA, ACBFAAP, ACBFXAA, and ACBFXAP) or summary accounting data (SABFAAA, SABFAAP, SABFXAA, and SABFXAP). It can be plan-oriented (ACBFAAP, SABFAAP, ACBFXAP, and SABFXAP) or user-oriented (ACBFAAA, SABFAAA, ACBFXAA, and SABFXAA).

```
BMC SOFTWARE INC                     DB2 ACCOUNTING BUFFER POOL(32K) REPORT - AVERAGES                            PAGE -     1
REPORT: ACBFAAA                                                                               REPORT DATE: 2001-03-25 13.42.05
DATE FROM: 2001-03-01 10.27.22                                                                               LOCATION - SANJOSE
DATE TO:   2001-03-19 16.55.20                                                                              SUBSYSTEM - DB2F

                         AVG     AVG     AVG     < PREFETCH REQUESTS >   AVG     AVG     < HP READS >    AVG    < HP WRITES >
                 AVG     SYNC    PAGE    SYNC   AVG     AVG     AVG     ASYNC   GETPAGE  AVG     AVG    HP->VP   AVG     AVG
AUTHID  #OCCUR  GETPAGE READ IO UPDATES WRT IO  SEQ.    LIST    DYNAMIC PG READ FAILED  REQUEST FAILED  PAGES   REQUEST FAILED
------- ------- ------- ------- ------- ------- ------- ------- ------- ------- ------- ------- ------- ------- ------- -------
CJN1          1     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0
CJN4          4     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0
MXW2         22    32.3     1.5    18.4     1.4     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0
MXW4          1     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0
NHJ1         15     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0
NHJ2          6     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0
NJ1X          1     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0
```

Figure 47. DB2 Accounting Buffer Pool Report—Averages

AUTHID
Authorization ID of the user accessing this DB2 system. This field is produced by ACBFpAA and SABFpAA.

PLANNAME
Plan name of the application program, transaction, or utility executed. This field is produced by ACBFpAP and SABFpAP.

#OCCUR
Number of times the plan was executed or number of plan executions for the user.

AVG GETPAGE
Average number of GETPAGE requests issued. This includes conditional and non-conditional, successful and unsuccessful requests.

AVG SYNC READ IO
Average number of synchronous read I/Os.

AVG PAGE UPDATES
Average number of times a buffer update occurs for the agent.

AVG SYNC WRT IO
Average number of immediate (synchronous) write I/Os.

PREFETCH REQUESTS AVG SEQ.
Average number of sequential prefetch requests.

PREFETCH REQUESTS AVG LIST
Average number of list prefetch requests.

PREFETCH REQUESTS AVG DYNAMIC
Average number of dynamic prefetch requests.

AVG ASYNC PG READ
Average number of asynchronous pages read by prefetch under the control of the agent.

AVG GETPAGE FAILED
Average number of times conditional GETPAGE requests could not be satisfied for this buffer pool.

HP READS AVG REQUEST
Average number of successful hiperpool reads.

HP READS AVG FAILED
Average number of unsuccessful hiperpool reads.

AVG HP->VP PAGES
Average number of pages that were found in a hiperpool and moved to a virtual buffer pool because of a prefetch under the control of the agent.

HP WRITES AVG REQUEST
Average number of successful hiperpool writes.

HP WRITES AVG FAILED
Average number of unsuccessful hiperpool writes.

## xxRIDy—DB2 Accounting RID List Access Report

The DB2 Accounting RID List Access Report provides an overview of RID (Record ID) pool usage. It is produced from either detailed accounting data (ACRIDA and ACRIDP) or summary accounting data (SARIDA and SARIDP). It can be plan-oriented (ACRIDP and SARIDP) or user-oriented (ACRIDA and SARIDA).

```
BMC SOFTWARE INC                             DB2 ACCOUNTING RID LIST ACCESS                                     PAGE 0001
REPORT: ACRIDA                                                                            REPORT DATE: 2001-03-25 10.25.45
DATE FROM: 2001-03-01  09:13:36                                                                         LOCATION - SANJOSE
DATE TO:   2001-03-19  11:13:17                                                                         SUBSYSTEM - DB2D

                            INSERT/            MIAP    MIAP
                            UPDATE/   INDEX     NO     OVER
AUTHID     #OCCUR  SELECTS  DELETE    PATHS   STORAGE   MAX
-------- -------- -------- -------- -------- -------- --------
ACCTDEP         5      454        0      214        0        0
BABUSER        42      225        0      101        0        0
CD10            5      168        0        2        0        0
CIM5           36        1        0        0        0        0
```

Figure 48. DB2 Accounting RID List Access Report

AUTHID
: Authorization ID of the user submitting this application. This field is produced by ACRIDA and SARIDA.

PLANNAME
: Plan name of the application program, transaction, or utility executed. This field is produced by ACRIDP and SARIDP.

# OCCUR
: Number of times the plan was executed or number of plan executions for the user.

SELECTS
: Number of SELECT statements processed for each plan or user.

INSERT/UPDATE/DELETE
: Number of INSERT, UPDATE, and DELETE statements processed for each plan or user.

INDEX PATHS
: Number of times RID list processing was used by each plan or user. This count includes all RID list processing, both for a single index (list prefetch) and multiple index access path.

MIAP NO STORAGE
: Total number of times RID list processing was not used by each plan or user because of insufficient storage in the RID pool.

MIAP OVER MAX
: Total number of times RID list processing was not used by each plan or user because the number of RIDs retrieved exceeded one or more internal limits, such as the maximum number of RIDs.

# xxGBATy—DB2 Accounting Global BPool Report—Totals

The DB2 Accounting Global BPool Report provides information about the total activity of all global buffer pools.

It is produced from either detailed accounting data (ACGBATA and ACGBATP) or summary accounting data (SAGBATA and SAGBATP). It can be plan-oriented (ACGBATP and SAGBATP) or user-oriented (ACGBATA and SAGBATA).

```
BMC SOFTWARE INC              DB2 ACCOUNTING GLOBAL BPOOL (ALL) REPORT - TOTALS                                     PAGE -     4
REPORT: ACGBATA                                                                              REPORT DATE: 2001-03-25 14.48.55
DATE FROM: 2001-03-01  10.25.27                                                                             LOCATION - SANJOSE
DATE TO:   2001-03-19  21.44.17                                                                             SUBSYSTEM - DB2G

                <----COUPLING FACILITY READ REQUESTS---->
                <-INVALID BUFFER---> <-PAGE NOT IN BPOOL> <-GRP BPOOL->
                 DATA  <--NO DATA-->  DATA  <--NO DATA--> <PGS WRITTEN> UNREG  EXPLCT WRITE  DEPEND <-ASYNC REQ-> <-PG PLOCK-->
AUTHID   #OCCUR RETURN RW INT NO INT RETURN RW INT NO INT CHANGE CLEAN  PG REQ X-INV  SECGBP GETPGS PRIMRY SECDRY  LOCK  SUSPND
-------- ------ ------ ------ ------ ------ ------ ------ ------ ------ ------ ------ ------ ------ ------ ------ ------ ------
BMCDWW1      76      0      0      0      0      6      0     12      0      0      0      0      0      0      0      0      0
BMCDWW2       5      0      0      0      0      0      0      1      0      0      0      0      0      0      0      0      0
BOLCJN2      15      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0
BOLCJN3      14      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0
BOLCJN4       2      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0
BOLHHH2       3      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0
BOLHHH3       2      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0
BOLJGQ2      84      6      1      0      6      6      0    177      0     71      0      0      0      0      0      0      0
BOLKGB1       1      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0
BOLSTC       37      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0
BOLWLB2       1      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0
BOLWTN1       6      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0
Q300         17      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0
```

Figure 49. DB2 Accounting Global BPool Report—Totals

AUTHID
Authorization ID of the user accessing this DB2 system. This field is produced by ACGBpAA and SAGBpAA.

PLANNAME
Plan name of the application program, transaction, or utility executed. This field is produced by ACGBpAP and SAGBpAP.

#OCCUR
Number of times the plan was executed or number of plan executions for the user.

COUPLING FACILITY READ REQUESTS-INVALID BUFFER-DATA RETURN
Total number of coupling facility read requests required because the buffer was marked invalid. Data was returned from the group buffer pool.

COUPLING FACILITY READ REQUESTS-INVALID BUFFER-NO DATA-RW INT
Total number of coupling facility read requests required because the buffer was marked invalid. Data is not returned from the group buffer pool, and a directory entry is created if it does not already exist. This means that another DB2 in the group has R/W interest in the page set or partition.

COUPLING FACILITY READ REQUESTS-INVALID BUFFER-NO DATA-NO INT
Total number of coupling facility read requests required because the buffer was marked invalid. Data is not returned from the group buffer pool, and no directory entry is created for this page. When no other DB2 in the group has R/W interest in the page set or partition, the process of creating the directory entry can be avoided.

COUPLING FACILITY READ REQUESTS-PAGE NOT IN BPOOL-DATA RETURN
Total number of coupling facility read requests required because the requested page was not found in the buffer pool. Data was returned from the coupling facility.

COUPLING FACILITY READ REQUESTS-PAGE NOT IN BPOOL-NO DATA-RW INT
Total number of coupling facility read requests required because the requested page was not found in the buffer pool. Data is not returned from the group buffer pool, and a directory entry is created if it does not already exist. This means that another DB2 in the group has R/W interest in the page set or partition.

COUPLING FACILITY READ REQUESTS-PAGE NOT IN BPOOL-NO DATA-NO INT
Total number of coupling facility read requests required because the requested page was not found in the buffer pool. Data is not returned from the group buffer pool, and no directory entry is created for this page. When no other DB2 in the group has R/W interest in the page set or partition, the process of creating the directory entry can be avoided.

GRP BPOOL-PGS WRITTEN-CHANGE
Total number of changed pages written to the group buffer pool.

GRP BPOOL-PGS WRITTEN-CLEAN
Total number of clean pages written to the group buffer pool. DB2 writes clean pages for page sets and partitions defined with GBPCACHE=ALL.

UNREG PG REQ
Number of coupling facility requests to unregister a page (DB2 5.1 and later only).

EXPLCT X-INV
Number of explicit cross-invalidations (DB2 6.1 and later only).

WRITE SECGBP
Number of coupling facility requests to write changed pages to the secondary group buffer pool for duplexing (DB2 6.1 and later only).

DEPEND GETPGS
Number of getpages for global buffer pool dependent pages (DB2 7.1 and later only).

ASYNC REQS-PRIMRY
Number of asynchronous IXLCACHE invocations for the primary group buffer pool (DB2 6.1 and later only).

ASYNC REQS-SECDRY
Number of asynchronous IXLCACHE invocations for the secondary group buffer pool (DB2 6.1 and later only).

PG PLOCK-LOCK
Number of page P-lock lock requests for space map, data, and index leaf pages (DB2 7.1 and later only).

PG PLOCK-SUSPND
Number of page P-lock lock suspensions for space map, data, and index leaf pages (DB2 7.1 and later only).

# xxGBAAy—DB2 Accounting Global BPool Report—Averages

The DB2 Accounting Global BPool Report provides information about the average activity of all global buffer pools.

It is produced from either detailed accounting data (ACGBAAA and ACGBAAP) or summary accounting data (SAGBAAA and SAGBAAP). It can be plan-oriented (ACGBAAP and SAGBAAP) or user-oriented (ACGBAAA and SAGBAAA).

```
BMC SOFTWARE INC               DB2 ACCOUNTING GLOBAL BPOOL (ALL) REPORT - AVERAGES                                 PAGE -     4
REPORT: ACGBAAA                                                                                  REPORT DATE: 2001-03-25 14.48.55
DATE FROM: 2001-03-01  10.25.27                                                                                LOCATION - SANJOSE
DATE TO:   2001-03-19  21.44.17                                                                                 SUBSYSTEM - DB2G

                <----COUPLING FACILITY READ REQUESTS---->
                <-INVALID BUFFER---> <-PAGE NOT IN BPOOL> <-GRP BPOOL->
                 DATA  <--NO DATA-->  DATA  <--NO DATA--> <PGS WRITTEN> UNREG  EXPLCT WRITE  DEPEND <-ASYNC REQ-> <-PG PLOCK-->
AUTHID   #OCCUR RETURN RW INT NO INT RETURN RW INT NO INT CHANGE CLEAN  PG REQ X-INV  SECGBP GETPGS PRIMRY SECDRY  LOCK  SUSPND
-------- ------ ------ ------ ------ ------ ------ ------ ------ ------ ------ ------ ------ ------ ------ ------ ------ ------
BMCDWW1      76    0.0    0.0    0.0    0.0    0.0    0.0    0.1    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0
BMCDWW2       5    0.0    0.0    0.0    0.0    0.0    0.0    0.2    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0
BOLCJN2      15    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0
BOLCJN3      14    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0
BOLCJN4       2    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0
BOLHHH2       3    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0
BOLHHH3       2    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0
BOLJGQ2      84    0.0    0.0    0.0    0.0    0.0    0.0    2.1    0.0    0.8    0.0    0.0    0.0    0.0    0.0    0.0    0.0
BOLKGB1       1    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0
BOLSTC       37    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0
BOLWLB2       1    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0
BOLWTN1       6    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0
Q300         17    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0
```

Figure 50. DB2 Accounting Global BPool Report—Averages

AUTHID
: Authorization ID of the user accessing this DB2 system. This field is produced by ACGBpAA and SAGBpAA.

PLANNAME
: Plan name of the application program, transaction, or utility executed. This field is produced by ACGBpAP and SAGBpAP.

#OCCUR
: Number of times the plan was executed or number of plan executions for the user.

COUPLING FACILITY READ REQUESTS-INVALID BUFFER-DATA RETURN
: Average number of coupling facility read requests required because the buffer was marked invalid. Data was returned from the group buffer pool.

COUPLING FACILITY READ REQUESTS-INVALID BUFFER-NO DATA-RW INT
: Average number of coupling facility read requests required because the buffer was marked invalid. Data is not returned from the group buffer pool, and a directory entry is created if it does not already exist. This means that another DB2 in the group has R/W interest in the page set or partition.

COUPLING FACILITY READ REQUESTS-INVALID BUFFER-NO DATA-NO INT
: Average number of coupling facility read requests required because the buffer was marked invalid. Data is not returned from the group buffer pool, and no directory entry is created for this page. When no other DB2 in the group has R/W interest in the page set or partition, the process of creating the directory entry can be avoided.

COUPLING FACILITY READ REQUESTS-PAGE NOT IN BPOOL-DATA RETURN
Average number of coupling facility read requests required because the requested page was not found in the buffer pool. Data was returned from the coupling facility.

COUPLING FACILITY READ REQUESTS-PAGE NOT IN BPOOL-NO DATA-RW INT
Average number of coupling facility read requests required because the requested page was not found in the buffer pool. Data is not returned from the group buffer pool, and a directory entry is created if it does not already exist. This means that another DB2 in the group has R/W interest in the page set or partition.

COUPLING FACILITY READ REQUESTS-PAGE NOT IN BPOOL-NO DATA-NO INT
Average number of coupling facility read requests required because the requested page was not found in the buffer pool. Data is not returned from the group buffer pool, and no directory entry is created for this page. When no other DB2 in the group has R/W interest in the page set or partition, the process of creating the directory entry can be avoided.

GRP BPOOL-PGS WRITTEN-CHANGE
Average number of changed pages written to the group buffer pool.

GRP BPOOL-PGS WRITTEN-CLEAN
Average number of clean pages written to the group buffer pool. DB2 writes clean pages for page sets and partitions defined with GBPCACHE=ALL.

UNREG PG REQ
Number of coupling facility requests to unregister a page (DB2 5.1 and later only).

EXPLCT X-INV
Number of explicit cross-invalidations (DB2 6.1 and later only).

WRITE SECGBP
Number of coupling facility requests to write changed pages to the secondary group buffer pool for duplexing (DB2 6.1 and later only).

DEPEND GETPGS
Number of getpages for global buffer pool dependent pages (DB2 7.1 and later only).

ASYNC REQS-PRIMRY
Number of asynchronous IXLCACHE invocations for the primary group buffer pool (DB2 6.1 and later only).

ASYNC REQS-SECDRY
Number of asynchronous IXLCACHE invocations for the secondary group buffer pool (DB2 6.1 and later only).

PG PLOCK-LOCK
Number of page P-lock lock requests for space map, data, and index leaf pages (DB2 7.1 and later only).

PG PLOCK-SUSPND
Number of page P-lock lock suspensions for space map, data, and index leaf pages (DB2 7.1 and later only).

# Chapter 14. Statistics Reports

This chapter describes the statistics reports produced by Performance Reporter, which include

- General DB2 system reports
  - The DB2 Statistics Overview Report—Averages
    - STOVER
  - The DB2 Statistics Overview Report—Totals
    - STOVERT
  - The DB2 Statistics Overview Report—DDF Averages
    - STOVR2
  - The DB2 Statistics Overview Report—DDF Totals
    - STOVR2T
  - The DB2 Statistics Thread-Related Exceptions Report
    - STEXTHD
  - The DB2 Statistics System Exceptions Report
    - STEXSYS
- User activity reports
  - The DB2 Statistics SQL Report—Averages
    - STSQL
  - The DB2 Statistics Dynamic SQL Statement Cache Report
    - STDYNC
  - The DB2 Statistics CPU Time Report
    - STCPU
  - DB2 Statistics Parallelism Report
    - STPLIO
  - DB2 Statistics Stored Procedures Report
    - STSPROC
  - The DB2 Statistics DDF Overview Report—Totals
    - STOVDFT

- Lock reports
  - The DB2 Statistics Lock Report
    - STLOCK
  - The DB2 Statistics Global Lock Report
    - STGLOCK
- EDM pool reports
  - The DB2 Statistics EDM Pool Report
    - STEDM
  - The DB2 Statistics EDM Pool Extended Report
    - STEDM2
- Buffer pool reports
  - The DB2 Statistics Buffer Pool Activity Report
    - STBFAVP
    - STBFXVP
  - The DB2 Statistics Buffer Pool Prefetch Activity Report
    - STBFAPF
    - STBFXPF
  - The DB2 Statistics Hiperpool Activity Report
    - STBFAHP
    - STBFXHP
  - The DB2 Statistics Buffer Pool Exceptions Report
    - STBFAEX
    - STBFXEX
  - The DB2 Statistics RID Activity Report
    - STRID
  - The DB2 Statistics Global BPool Activity Report
    - STGBA
    - STGBX
  - The DB2 Statistics Global BPool Activity Extended Report
    - STGBA2
    - STGBX2
  - The DB2 Statistics Global BPool Exceptions Report
    - STGBAEX
    - STGBXEX
- Log report
  - The DB2 Statistics Logging Report
    - STLOG

## General DB2 System Reports

For a statistics interval, the general DB2 system reports provide averages or totals about the amount of activity occurring within a DB2 system operating period, including DDF averages and totals, and thread- or system-related problems.

## STOVER—DB2 Statistics Overview Report—Averages

The DB2 Statistics Overview Report contains overview indicators of the status of the system through its operating period. This report lists the average number of occurrences of these indicators per create thread during each statistics interval.

```
BMC SOFTWARE INC                               DB2 STATISTICS OVERVIEW REPORT - AVERAGES                              PAGE 0002
REPORT: STOVER                                                                                 REPORT DATE: 2001-03-25 11.14.54
DATE FROM: 2001-03-01 01:39:01                                                                                 LOCATION - SANJOSE
DATE TO:   2001-03-19 03:03:49                                                                                SUBSYSTEM - DB2P

                      CREATE PHASE 2   SYNC           SQL     SQL     SQL   GETPAGES PAGE   READ   WRITE    EDM LOG WRT    LOCK
 DATETIME            THREADS COMMITS COMMITS ABORTS  MANIP. CONTROL DEFINIT   REQS  UPDATES I/O      I/O   LOADS    CALLS SUSPEND
 ------------------- ------- ------- ------- ------ ------- ------- ------- ------- ------- ----- ------- ------- ------- -------
 2001-03-01-06.50.23       1     0.0     0.0    0.0     0.0     0.0     0.0     6.0     0.0   1.0     0.0     0.0     0.0     0.0
 2001-03-01-07.20.18       0     0.0     0.0    0.0     0.0     0.0     0.0     0.0     0.0   0.0     0.0     0.0     0.0     0.0
 2001-03-01-07.50.11       8     0.6     0.5    0.1     1.2     2.2     0.0   367.7   176.7  15.7     1.3     1.3     4.0     0.0
 2001-03-01-08.20.05       3     0.3     1.0    0.0     0.6     0.0     0.0    64.6    30.6   1.6     4.0     0.0     2.0     0.0
 2001-03-01-08.50.01       1     0.0     1.0    0.0   253.0     0.0     0.0   520.0     0.0  15.0     0.0     2.0     0.0     0.0
 2001-03-01-09.19.55       5     0.0     2.8    0.0     0.0     9.2     2.6      13   671.8  13.4     7.8     1.4    12.8     0.0
 2001-03-01-09.49.53       0     0.0     0.0    0.0     0.0     0.0     0.0     0.0     0.0   0.0     0.0     0.0     0.0     0.0
 2001-03-01-10.19.45       4     0.0     1.2    0.0     0.7     1.2     0.0   759.7   358.0   4.2     5.7     1.2     6.7     0.0
 2001-03-01-10.49.36      16     1.0     0.0    0.0     1.0     0.0     0.0     4.8     0.0   3.7     0.0     0.0     0.0     0.0
 2001-03-01-11.19.34       1     2.0     0.0    0.0     1.0     0.0     0.0    10.0     0.0   8.0     0.0     0.0     0.0     0.0
 2001-03-01-11.49.30       5     0.6     0.2    0.0     0.6     0.4     0.0    15.8     0.0   3.2     0.0     0.0     0.0     0.0
 2001-03-01-12.19.20      14     1.0     0.0    0.0     1.0     0.8     0.0    17.3     0.0   4.8     0.0     0.2     0.0     0.0
 2001-03-01-12.49.20       2     0.0     0.5    0.0     0.0     0.0     0.0     3.0     0.0   0.0     0.0     0.0     0.0     0.0
 2001-03-01-13.19.42       1     0.0     1.0    0.0     0.0     0.0     0.0     4.0     0.0   0.0     0.0     0.0     0.0     0.0
 2001-03-01-13.49.29       4     0.7     0.0    0.2     3.2     0.5     0.0    20.2     3.5   3.7     0.0     2.2     1.7     0.0
 2001-03-01-14.19.24       1     0.0     1.0    0.0     0.0     0.0     0.0      15      11     2    18.0     1.0    28.0     0.0
 2001-03-01-14.49.21      14     0.7     0.5    0.0     1.9     2.8     0.0    36.8     5.5   5.6     0.2     0.3     0.6     0.0
 2001-03-01-15.19.12      10     0.6     0.7    0.0     1.4     1.5     0.0   304.6   168.1   4.7     2.4     0.0     3.4     0.0
```

Figure 51. DB2 Statistics Overview Report—Averages

DATETIME
: Date and time that this statistics interval ended.

CREATE THREADS
: Number of threads created in this statistics interval.

PHASE 2 COMMITS
: Average number of phase 2 commits per CREATE THREAD during this statistics interval.

SYNC COMMITS
: Average number of synchronized commit requests per CREATE THREAD during this statistics interval. Synchronized commit is used by TSO applications, the Call Attach Facility, and batch applications.

ABORTS
: Average number of aborts per CREATE THREAD during this statistics interval.

SQL MANIP.
: Average number of SQL data manipulation statements executed per CREATE THREAD during this statistics interval.

SQL CONTROL
Average number of SQL CONTROL statements executed per CREATE THREAD during this statistics interval.

SQL DEFINIT
Average number of SQL data definition statements executed per CREATE THREAD during this statistics interval.

GETPAGE REQS
Average number of GETPAGE requests per CREATE THREAD during this statistics interval.

PAGE UPDATES
Average number of rows updated in system pages per CREATE THREAD during this statistics interval.

READ I/O
Average number of read I/Os issued per CREATE THREAD during this statistics interval. This includes both synchronous and asynchronous read I/Os.

WRITE I/O
Average number of write I/Os issued per CREATE THREAD during this statistics interval. This includes both synchronous and asynchronous write I/Os.

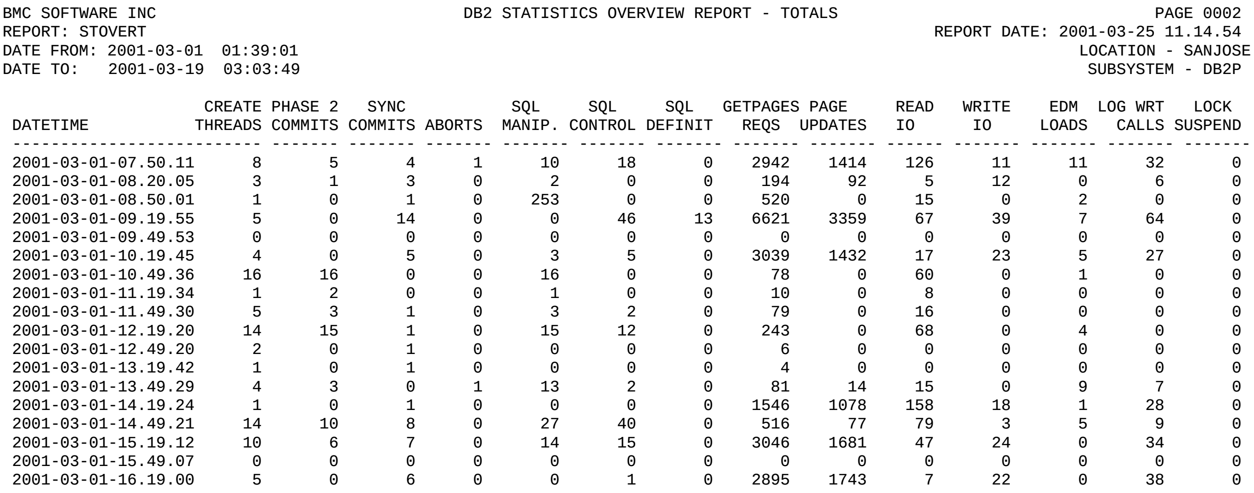
EDM LOADS
Average number of EDM pool loads from the DB2 directory per CREATE THREAD during this statistics interval.

LOG WRT CALLS
Average number of write requests to the active log per CREATE THREAD during this statistics interval.

LOCK SUSPEND
Average number of suspensions due to waiting for lock per CREATE THREAD during this statistics interval.

# STOVERT—DB2 Statistics Overview Report—Totals

The DB2 Statistics Overview Report contains overview indicators of the status of the system through its operating period. This report lists the total number of occurrences of these indicators per CREATE THREAD during each statistics interval.



```
BMC SOFTWARE INC                                DB2 STATISTICS OVERVIEW REPORT - TOTALS                                 PAGE 0002
REPORT: STOVERT                                                                                  REPORT DATE: 2001-03-25 11.14.54
DATE FROM: 2001-03-01  01:39:01                                                                                LOCATION - SANJOSE
DATE TO:   2001-03-19  03:03:49                                                                                 SUBSYSTEM - DB2P

                     CREATE PHASE 2   SYNC           SQL     SQL     SQL   GETPAGES PAGE     READ   WRITE    EDM  LOG WRT   LOCK
 DATETIME           THREADS COMMITS COMMITS ABORTS  MANIP. CONTROL DEFINIT   REQS  UPDATES   IO      IO    LOADS   CALLS SUSPEND
 ------------------------- ------- ------- ------- ------- ------- ------- ------- ------- ------ ------- ------- ------- -------
 2001-03-01-07.50.11      8       5       4      1      10      18       0    2942    1414    126      11      11      32       0
 2001-03-01-08.20.05      3       1       3      0       2       0       0     194      92      5      12       0       6       0
 2001-03-01-08.50.01      1       0       1      0     253       0       0     520       0     15       0       2       0       0
 2001-03-01-09.19.55      5       0      14      0       0      46      13    6621    3359     67      39       7      64       0
 2001-03-01-09.49.53      0       0       0      0       0       0       0       0       0      0       0       0       0       0
 2001-03-01-10.19.45      4       0       5      0       3       5       0    3039    1432     17      23       5      27       0
 2001-03-01-10.49.36     16      16       0      0      16       0       0      78       0     60       0       1       0       0
 2001-03-01-11.19.34      1       2       0      0       1       0       0      10       0      8       0       0       0       0
 2001-03-01-11.49.30      5       3       1      0       3       2       0      79       0     16       0       0       0       0
 2001-03-01-12.19.20     14      15       1      0      15      12       0     243       0     68       0       4       0       0
 2001-03-01-12.49.20      2       0       1      0       0       0       0       6       0      0       0       0       0       0
 2001-03-01-13.19.42      1       0       1      0       0       0       0       4       0      0       0       0       0       0
 2001-03-01-13.49.29      4       3       0      1      13       2       0      81      14     15       0       9       7       0
 2001-03-01-14.19.24      1       0       1      0       0       0       0    1546    1078    158      18       1      28       0
 2001-03-01-14.49.21     14      10       8      0      27      40       0     516      77     79       3       5       9       0
 2001-03-01-15.19.12     10       6       7      0      14      15       0    3046    1681     47      24       0      34       0
 2001-03-01-15.49.07      0       0       0      0       0       0       0       0       0      0       0       0       0       0
 2001-03-01-16.19.00      5       0       6      0       0       1       0    2895    1743      7      22       0      38       0
```

Figure 52. DB2 Statistics Overview Report—Totals

DATETIME
: Date and time that this statistics interval ended.

CREATE THREADS
: Number of threads created in this statistics interval.

PHASE 2 COMMITS
: Total number of phase 2 commits per CREATE THREAD during this statistics interval.

SYNC COMMITS
: Total number of synchronous commits per CREATE THREAD during this statistics interval.

ABORTS
: Total number of aborts per CREATE THREAD during this statistics interval.

SQL MANIP.
: Total number of SQL data manipulation statements executed per CREATE THREAD during this statistics interval.

SQL CONTROL
: Total number of SQL CONTROL statements executed per CREATE THREAD during this statistics interval.

SQL DEFINIT
: Total number of SQL data definition statements executed per CREATE THREAD during this statistics interval.

GETPAGE REQS

Total number of GETPAGE requests per CREATE THREAD during this statistics interval.

PAGE UPDATES

Total number of rows updated in system pages per CREATE THREAD during this statistics interval.

READ I/O

Total number of read I/Os issued per CREATE THREAD during this statistics interval. This includes both synchronous and asynchronous read I/Os.

WRITE I/O

Total number of write I/Os issued per CREATE THREAD during this statistics interval. This includes both synchronous and asynchronous write I/Os.

EDM LOADS

Total number of EDM pool loads from the DB2 directory per CREATE THREAD during this statistics interval.

LOG WRT CALLS

Total number of write requests to the active log per CREATE THREAD during this statistics interval.

LOCK SUSPEND

Total number of suspensions due to waiting for lock per CREATE THREAD during this statistics interval.

## STEXTHD—DB2 Statistics Thread-Related Exceptions Report

The DB2 Statistics Thread-Related Exceptions Report displays a row of information per statistics interval if one or more of the following indicators of thread-related performance problems is greater than zero.

```
BMC SOFTWARE INC                        DB2 STATISTICS THREAD-RELATED EXCEPTIONS REPORT                                PAGE 0001
REPORT: STEXTHD                                                                                         REPORT DATE: 2001-03-25 11.14.54
DATE FROM: 2001-03-01  11.47.44                                                                                    LOCATION - SANJOSE
DATE TO:   2001-03-19  03:03:49                                                                                     SUBSYSTEM - DB2C

                     CREATE                                     AUTH   AUTO   AUTO         RESOLVE                     DISTR.
                     THREAD END OF END OF                       CHECK  BINDS  BINDS THREAD THREAD  LOCK  INVALID DBAT  CONVS
 DATETIME            WAITS  TASK   MEMORY DEADLK TMEOUT BCKOUT FAILED   OK   FAILED INDBT  INDBT  ESCALA. COL.   QUEUED DEALLOC
 ------------------- ------ ------ ------ ------ ------ ------ ------  -----  ------ ------ ------ ------ ------ ------ -------
 2001-03-01-11.47.44      0      0      0      0      0      0      0      1      0      0      0      0      0      0       0
 2001-03-01-00.22.07      0      3      0      0      0      0      0      0      0      0      0      0      0      0       0
 2001-03-01-11.33.52      0      0      0      0      0      0      0      1      0      0      0      0      0      0       0
 2001-03-01-17.01.55      0      0      0      0      0      2      0      0      0      0      0      0      0      0       0
 2001-03-01-18.06.15      0      4      0      0      0      0      0      0      0      0      0      0      0      0       0
```

Figure 53. DB2 Statistics Thread-Related Exceptions Report

DATETIME
: Date and time that this statistics interval ended.

CREATE THREAD WAITS
: Number of CREATE THREAD requests that waited for a thread during this statistics interval.

END OF TASK
: Number of end of task failures during this statistics interval. This counter is incremented when an end of task request is processed for a DB2-connected address space; for example, an abend of an IMS/VS or CICS task having a DB2 thread.

END OF MEMORY
: Number of end of memory failures during this statistics interval. This counter is incremented when an end of memory request is processed for a DB2-connected address space; for example, at TSO force.

DEADLK
: Number of deadlocks detected during this statistics interval.

TMEOUT
: Number of lock timeouts during this statistics interval.

BCKOUT
: Number of successful aborts; the number of times a unit of recovery was backed out due to an application program abend or rollback request, a deadlock or timeout because of database locking, an application cancelled by the operator, or a thread abended because of resource shortage.

AUTH CHECK FAILED
: Number of times an authorization check failed during this statistics interval.

AUTO BINDS OK
: Number of automatic binds completed successfully during this statistics interval.

AUTO BINDS FAILED
Number of automatic binds attempted but failed during this statistics interval.

THREAD INDBT
Number of indoubt threads during this statistics interval.

RESOLVE THREAD INDBT
Number of successful indoubt unit of recovery requests.

LOCK ESCALA.
Number of lock escalations during this statistics interval.

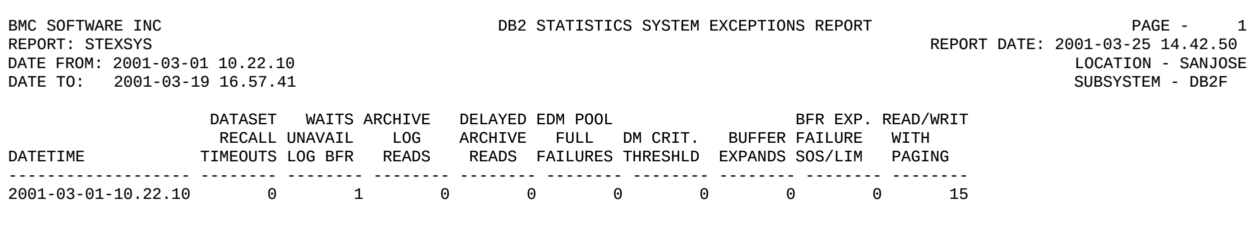
INVALID COL.
Total number of columns (rows times columns) for which an invalid select procedure was encountered during this statistics interval.

DBAT QUEUED
Number of times a database access thread was queued because the maximum number of active remote threads was reached.

DISTR CONVS DEALLOC
Number of distributed conversations deallocated because the maximum number of concurrent (active plus inactive) threads was reached.

## STEXSYS—DB2 Statistics System Exceptions Report

The DB2 Statistics System Exceptions Report displays a row of information per statistics interval if one or more of the following indicators of system performance problems is greater than zero.



```
BMC SOFTWARE INC                                   DB2 STATISTICS SYSTEM EXCEPTIONS REPORT                            PAGE -     1
REPORT: STEXSYS                                                                                 REPORT DATE: 2001-03-25 14.42.50
DATE FROM: 2001-03-01 10.22.10                                                                                  LOCATION - SANJOSE
DATE TO:   2001-03-19 16.57.41                                                                                 SUBSYSTEM - DB2F

                     DATASET   WAITS ARCHIVE   DELAYED EDM POOL                   BFR EXP. READ/WRIT
                      RECALL UNAVAIL    LOG    ARCHIVE   FULL   DM CRIT.   BUFFER FAILURE   WITH
DATETIME            TIMEOUTS LOG BFR   READS    READS  FAILURES THRESHLD  EXPANDS SOS/LIM   PAGING
------------------- -------- -------- -------- -------- -------- -------- -------- -------- --------
2001-03-01-10.22.10        0        1        0        0        0        0        0        0       15
```

Figure 54. DB2 Statistics System Exceptions Report

DATETIME
Date and time that this statistics interval ended.

DATASET RECALL TIMEOUTS
Number of times HSM data set recall did not complete within the time limit during this statistics interval.

WAITS UNAVAIL LOG BFR
Number of times logging activity had to wait because of unavailable output buffers during this statistics interval.

ARCHIVE LOG READS
Number of reads from an archive log to perform backouts during this statistics interval.

DELAYED ARCHIVE READS
Number of reads from an archive log delayed because of tape volume contention (DB2 3.1) or unavailable resources.

EDM POOL FULL FAILURES
Number of requests to DB2 to execute a plan that failed because of EDM out-of-space conditions during this statistics interval.

DM CRIT. THRESHLD
Number of times the Data Manager Critical threshold was reached during this statistics interval.

BUFFER EXPANDS
Number of buffer pool expansions requested during this statistics interval because an ALTER BUFFERPOOL command was issued.

BFR EXP. FAILURE SOS/LIM
Number of times a buffer pool expansion failed because there was not enough storage, or maximum buffer pool size was reached during this statistics interval. This may also be the number of times that a usable buffer could not be located because the pool was full.

READ/WRIT WITH PAGING
Number of page faults experienced when DB2 references a page before starting a read or write operation during this statistics interval.

# User Activity Reports

For a statistics interval, the user activity reports provide averages about the amount of SQL statements processed per CREATE THREAD, provide information about dynamic SQL statement cache activity, indicate the amount of CPU consumed, give information about parallel I/O activity, and summarize DDF activity.

## STSQL—DB2 Statistics SQL Report—Averages

The DB2 Statistics SQL Report shows the average number of each of the following types of SQL statements per CREATE THREAD during each statistics interval.

```
BMC SOFTWARE INC                                  DB2 STATISTICS SQL REPORT - AVERAGES                                  PAGE 0002
REPORT: STSQL                                                                                    REPORT DATE: 2001-03-25 11.14.54
DATE FROM: 2001-03-01  01:39:01                                                                                 LOCATION - SANJOSE
DATE TO:   2001-03-19  03:03:49                                                                                  SUBSYSTEM - DB2P

                      CREATE                                                                              OTHER
 DATETIME            THREADS  SELECT  INSERT  UPDATE  DELETE PREPARE   OPEN  FETCH  CREATE    DROP   ALTER   SQL     ABORT  COMMIT
 ------------------- ------- ------- ------- ------- ------- ------- ------ ------ ------- ------- ------- ------- ------- -------
 2001-03-01-07.50.11       8     0.6     0.0     0.6     0.0     0.0    1.1    4.1     0.0     0.0     0.0     0.1     0.1     0.6
 2001-03-01-08.20.05       3     0.3     0.0     0.3     0.0     0.0    0.0    0.0     0.0     0.0     0.0     0.3     0.0     0.3
 2001-03-01-08.50.01       1   253.0     0.0     0.0     0.0     0.0    0.0    0.0     0.0     0.0     0.0     0.0     0.0     0.0
 2001-03-01-09.19.55       5     0.0     0.0     0.0     0.0     4.6    0.0    0.0     2.0     0.6     0.0     5.6     0.0     0.0
 2001-03-01-09.49.53       0     0.0     0.0     0.0     0.0     0.0    0.0    0.0     0.0     0.0     0.0     0.0     0.0     0.0
 2001-03-01-10.19.45       4     0.0     0.7     0.0     0.0     0.7    0.2    0.5     0.0     0.0     0.0     0.2     0.0     0.0
 2001-03-01-10.49.36      16     1.0     0.0     0.0     0.0     0.0    0.0    0.0     0.0     0.0     0.0     0.0     0.0     1.0
 2001-03-01-11.19.34       1     1.0     0.0     0.0     0.0     0.0    0.0    0.0     0.0     0.0     0.0     0.0     0.0     2.0
 2001-03-01-11.49.30       5     0.6     0.0     0.0     0.0     0.0    0.2    0.4     0.0     0.0     0.0     0.0     0.0     0.6
 2001-03-01-12.19.20      14     1.0     0.0     0.0     0.0     0.0    0.4    3.7     0.0     0.0     0.0     0.0     0.0     1.0
 2001-03-01-12.49.20       2     0.0     0.0     0.0     0.0     0.0    0.0    0.0     0.0     0.0     0.0     0.0     0.0     0.0
 2001-03-01-13.19.42       1     0.0     0.0     0.0     0.0     0.0    0.0    0.0     0.0     0.0     0.0     0.0     0.0     0.0
 2001-03-01-13.49.29       4     2.5     0.2     0.0     0.5     0.0    0.2    1.2     0.0     0.0     0.0     0.0     0.2     0.7
 2001-03-01-14.19.24       1     0.0     0.0     0.0     0.0     0.0    0.0    0.0     0.0     0.0     0.0     0.0     0.0     0.0
 2001-03-01-14.49.21      14     1.1     0.5     0.2     0.0     1.1    0.8    7.2     0.0     0.0     0.0     0.0     0.0     0.7
 2001-03-01-15.19.12      10     0.9     0.5     0.0     0.0     0.5    0.5    5.9     0.0     0.0     0.0     0.0     0.0     0.6
 2001-03-01-15.49.07       0     0.0     0.0     0.0     0.0     0.0    0.0    0.0     0.0     0.0     0.0     0.0     0.0     0.0
 2001-03-01-16.19.00       5     0.0     0.0     0.0     0.0     0.2    0.0    0.0     0.0     0.0     0.0     0.2     0.0     0.0
```

Figure 55. DB2 Statistics SQL Report - Averages

DATETIME
: Date and time that this statistics interval ended.

CREATE THREADS
: Number of threads created in this statistics interval.

SELECT
: Average number of SELECT statements processed per CREATE THREAD during this statistics interval.

INSERT
: Average number of INSERT statements processed per CREATE THREAD during this statistics interval.

UPDATE
: Average number of UPDATE statements processed per CREATE THREAD during this statistics interval.

DELETE
: Average number of DELETE statements processed per CREATE THREAD during this statistics interval.

PREPARE
: Average number of PREPARE statements processed per CREATE THREAD during this statistics interval.

OPEN
: Average number of OPEN statements processed per CREATE THREAD during this statistics interval.

FETCH
: Average number of FETCH statements processed per CREATE THREAD during this statistics interval.

CREATE
: Average number of CREATE statements processed per CREATE THREAD during this statistics interval.

DROP
: Average number of DROP statements processed per CREATE THREAD during this statistics interval.

ALTER
: Average number of ALTER statements processed per CREATE THREAD during this statistics interval.

OTHER SQL
: Average number of other SQL statements processed per CREATE THREAD during this statistics interval.

ABORT
: Average number of unit of work aborts (roll-backs) performed per CREATE THREAD during this statistics interval.

COMMIT
: Average number of unit of work commits performed per CREATE THREAD during this statistics interval.

## STDYNC—DB2 Statistics Dynamic SQL Statement Cache Report

The DB2 Statistics Dynamic SQL Statement Cache Report provides information about dynamic SQL statement cache activity per CREATE THREAD during each statistics interval. This report is available for DB2 5.1 and later only.

```
BMC SOFTWARE INC                  DB2 STATISTICS DYNAMIC SQL STATEMENT CACHE REPORT                                 PAGE -     1
REPORT: STDYNC                                                                                  REPORT DATE: 2001-03-25 11.31.16
DATE FROM: 2001-03-01 14.05.10                                                                                 LOCATION - SANJOSE
DATE TO:   2001-03-19 01.01.39                                                                                 SUBSYSTEM - DB1G

                     CREATE    CACHE   CACHE    CACHE     STMT   STMT NOT  PREPARE IMPLICIT   STMT     STMT
DATETIME            THREADS    PAGES  INSERTS  REQUESTS  FOUND    FOUND    AVOIDED PREPARE  DISCARD   PURGED
------------------- -------- -------- -------- -------- -------- -------- -------- -------- -------- --------

2001-03-01-14.05.10        0        0        0        0        0        0        0        0        0        0
2001-03-01-14.19.27       10        0        0        0        0        0        0        0        0        0
2001-03-01-14.34.24        2        0        0        0        0        0        0        0        0        0
2001-03-01-14.49.22        1        0        0        0        0        0        0        0        0        0
2001-03-01-15.04.19        4        0        0        0        0        0        0        0        0        0
2001-03-01-15.19.17        0        0        0        0        0        0        0        0        0        0
2001-03-01-15.34.15        0        0        0        0        0        0        0        0        0        0
```

Figure 56. DB2 Statistics Dynamic SQL Statement Cache Report

DATETIME
Date and time that this statistics interval ended.

CREATE THREADS
Number of threads created in this statistics interval.

CACHE PAGES
Number of pages used for the dynamic statement cache.

CACHE INSERTS
Number of inserts into the dynamic statement cache.

CACHE REQUESTS
Number of requests to the dynamic statement cache.

STMT FOUND
Number of times DB2 satisfied a PREPARE request by making a copy of a statement in the prepared statement cache.

STMT NOT FOUND
Number of times DB2 searched the prepared statement cache but could not find a suitable prepared statement.

PREPARE AVOIDED
Number of times DB2 did not PREPARE a statement bound with KEEPDYNAMIC(YES) because the prepared statement cache contained a valid copy of the prepared statement.

IMPLICIT PREPARE
Number of times DB2 did an implicit PREPARE for a statement bound with KEEPDYNAMIC(YES) because the prepared statement cache did not contain a valid copy of the prepared statement.

STMT DISCARD
Number of times DB2 discarded a prepared statement from the prepared statement cache because the number of prepared statements in the cache exceeded the value of subsystem parameter MAXKEEPD.

STMT PURGED
Number of times DB2 discarded a prepared statement from the prepared statement cache because a program executed a DROP, ALTER, or REVOKE statement against a dependent object.

# STCPU—DB2 Statistics CPU Time Report

The DB2 Statistics CPU Time Report shows the amount of CPU consumed by DB2 processing during each statistics interval. Each value is displayed in seconds and is the sum of both TCB and SRB times.

```
BMC SOFTWARE INC                                  DB2 STATISTICS CPU TIME REPORT                                    PAGE -    10
REPORT: STCPU                                                                                   REPORT DATE: 2001-03-25 14.48.55
DATE FROM: 2001-03-01 10.36.22                                                                                 LOCATION - SANJOSE
DATE TO:   2001-03-19 21.41.03                                                                                 SUBSYSTEM - DB2G

                                     AVG                   SYSTEM    DATA BASE
                        CREATE    OF TOTAL       TOTAL    SERVICES   SERVICES      IRLM         DDF
DATETIME               THREADS       CPU          CPU        CPU        CPU         CPU         CPU
-------------------   -------- ----------- ----------- ----------- ----------- ----------- -----------
2001-03-01-10.36.22          8    1.074489    8.595917    2.947050    3.286774    2.330629    0.031464
2001-03-01-10.51.19          0    0.000000    5.703876    2.258580    1.294280    2.119196    0.031820
2001-03-01-11.06.17          0    0.000000    4.971499    2.234029    0.874565    1.834982    0.027923
2001-03-01-11.21.14          0    0.000000    5.025383    2.309152    0.845123    1.840585    0.030523
2001-03-01-11.36.12          2    3.336094    6.672189    2.674088    1.785634    2.183489    0.028978
2001-03-01-11.50.49          5    1.494492    7.472463    2.811381    2.324711    2.306819    0.029552
2001-03-01-12.06.07          9    1.803309   16.229788    2.915235   10.857058    2.421370    0.036125
2001-03-01-12.21.05         10    1.306173   13.061732    3.170725    7.390650    2.467457    0.032900
2001-03-01-12.34.43         11    0.958233   10.540568    2.628201    4.951550    2.933392    0.027425
2001-03-01-12.51.00         13    1.571451   20.428870    5.196157   11.911806    3.282113    0.038794
2001-03-01-13.05.58          0    0.000000    7.635387    2.769207    2.165906    2.669497    0.030777
2001-03-01-13.20.55          0    0.000000    5.185050    2.360681    0.918269    1.877026    0.029074
2001-03-01-13.35.53          0    0.000000    5.111885    2.368907    0.845416    1.866757    0.030805
2001-03-01-13.50.50          0    0.000000    5.002877    2.292855    0.846409    1.836194    0.027419
```

Figure 57. DB2 Statistics CPU Time Report

DATETIME
Date and time that this statistics interval ended.

CREATE THREADS
Number of threads created in this statistics interval.

AVG OF TOTAL CPU
Average number of CPU (TCB+SRB) seconds consumed by all active DB2 address spaces (system services, database services, and DDF) and the IRLM (IMS Resource Lock Manager) per CREATE THREAD during this statistics interval. This average is calculated from the TOTAL CPU field described below.

TOTAL CPU
Total number of CPU (TCB+SRB) seconds consumed by all active DB2 address spaces (system services, database services, and DDF) and the IRLM (IMS Resource Lock Manager) during this statistics interval.

SYSTEM SERVICES CPU
Total number of CPU (TCB+SRB) seconds consumed by the DB2 master address space during this statistics interval.

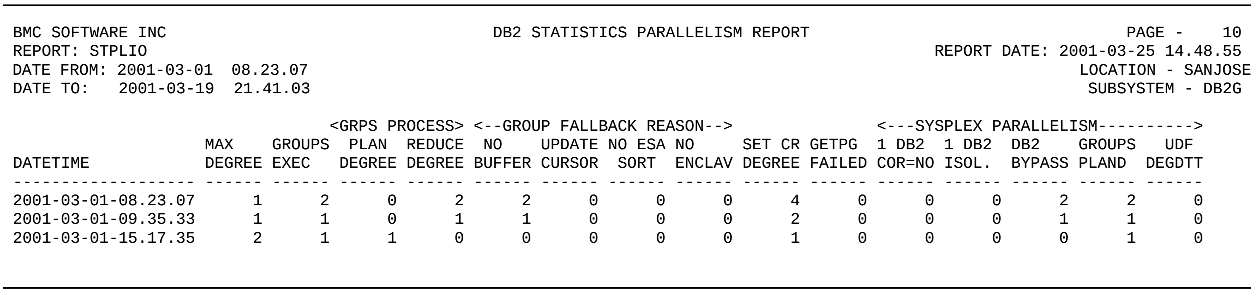
DATA BASE SERVICES CPU
Total number of CPU (TCB+SRB) seconds consumed by the DB2 database manager address space during this statistics interval.

IRLM CPU
Total number of CPU (TCB+SRB) seconds consumed by the IRLM address space during this statistics interval.

DDF CPU
Total number of CPU (TCB+SRB) seconds consumed by the DDF (DIST) address space if DDF is active during this statistics interval.

## STPLIO—DB2 Statistics Parallelism Report

The DB2 Statistics Parallelism Report provides information about parallel query activity, including both I/O and CPU parallelism, during each statistics interval.



```
BMC SOFTWARE INC                                  DB2 STATISTICS PARALLELISM REPORT                              PAGE -    10
REPORT: STPLIO                                                                                   REPORT DATE: 2001-03-25 14.48.55
DATE FROM: 2001-03-01  08.23.07                                                                                 LOCATION - SANJOSE
DATE TO:   2001-03-19  21.41.03                                                                                 SUBSYSTEM - DB2G

                                 <GRPS PROCESS> <--GROUP FALLBACK REASON-->               <---SYSPLEX PARALLELISM---------->
                    MAX    GROUPS  PLAN  REDUCE  NO    UPDATE NO ESA NO      SET CR GETPG  1 DB2  1 DB2  DB2    GROUPS   UDF
DATETIME            DEGREE EXEC   DEGREE DEGREE BUFFER CURSOR  SORT  ENCLAV DEGREE FAILED COR=NO ISOL.  BYPASS PLAND  DEGDTT
------------------- ------ ------ ------ ------ ------ ------ ------ ------ ------ ------ ------ ------ ------ ------ ------
2001-03-01-08.23.07      1      2      0      2      2      0      0      0      4      0      0      0      2      2      0
2001-03-01-09.35.33      1      1      0      1      1      0      0      0      2      0      0      0      1      1      0
2001-03-01-15.17.35      2      1      1      0      0      0      0      0      1      0      0      0      0      1      0
```

Figure 58.  DB2 Statistics Parallelism Report

DATETIME
: Date and time that this statistics interval ended.

MAX DEGREE
: Maximum degree of parallelism among the parallel groups to indicate the extent to which query parallelism applies during this statistics interval.

GROUPS EXEC
: Total number of parallel groups executed during this statistics interval.

GRPS PROCESS PLAN DEGREE
: Total number of parallel groups with a planned degree greater than one at run time, and were executed to the same degree because of sufficient storage on the buffer pool during this statistics interval.

GRPS PROCESS REDUCE DEGREE
: Total number of parallel groups with a planned degree greater than one at run time, but were processed to a parallel degree less than planned because of a storage shortage or contention on the buffer pool during this statistics interval.

GROUP FALLBACK REASON NO BUFFER
: Total number of parallel groups with a planned degree greater than one at run time, but fell back to sequential mode because of storage shortage or contention on the buffer pool during this statistics interval.

GROUP FALLBACK REASON UPDATE CURSOR
: Total number of parallel groups that fell back to sequential mode because the cursor may be used in UPDATE/DELETE during this statistics interval.

GROUP FALLBACK REASON NO ESA SORT
: Total number of parallel groups that fell back to sequential mode due to lack of ESA sort support during this statistics interval.

GROUP FALLBACK REASON NO ENCLAV
: Total number of parallel groups that fell back to sequential mode due to lack of MVS/ESA 5.2 SRB enclave services.

SET CR DEGREE

Number of SET CURRENT DEGREE statements executed during this statistics interval.

GETPG FAILED

Number of times conditional GETPAGE requests could not be satisfied for this buffer pool during this statistics interval.

SYSPLEX PARALLELISM 1 DB2 COR=NO

Number of parallel groups executed on a single DB2 because the coordinator value is set to NO (DB2 5.1 and later only).

SYSPLEX PARALLELISM 1 DB2 ISOL.

Total number of parallel groups executed on a single DB2 because the plan or package was bound with an isolation value of repeatable read (DB2 5.1 and later only).

SYSPLEX PARALLELISM DB2 BYPASS

Number of times the parallelism coordinator had to bypass a DB2 because of not enough buffer pool storage (DB2 5.1 and later only).

SYSPLEX PARALLELISM GROUPS PLAND

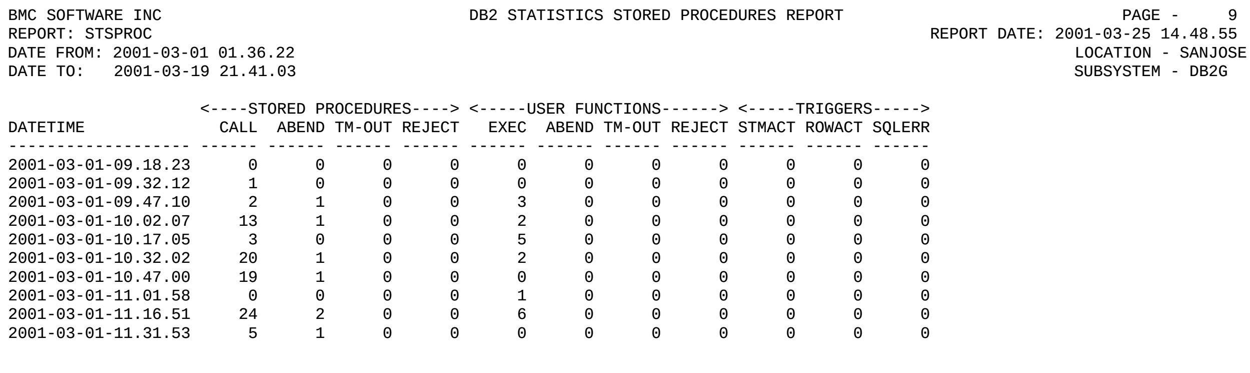
Number of parallel groups DB2 intended to run across the data sharing group (DB2 5.1 and later only).

SYSPLEX PARALLELISM UDF DEGDTT

Number of parallel groups that are part of a query block using a user-defined function with a Declared Temporary Table (DB2 7.1 and later only).

# STSPROC—DB2 Statistics Stored Procedures Report

The DB2 Statistics Stored Procedures Report provides information about stored procedure activity during each statistics interval.



```
BMC SOFTWARE INC                                DB2 STATISTICS STORED PROCEDURES REPORT                             PAGE -     9
REPORT: STSPROC                                                                                 REPORT DATE: 2001-03-25 14.48.55
DATE FROM: 2001-03-01 01.36.22                                                                                 LOCATION - SANJOSE
DATE TO:   2001-03-19 21.41.03                                                                                 SUBSYSTEM - DB2G

                    <----STORED PROCEDURES----> <-----USER FUNCTIONS------> <-----TRIGGERS----->
DATETIME              CALL  ABEND TM-OUT REJECT   EXEC  ABEND TM-OUT REJECT STMACT ROWACT SQLERR
------------------- ------ ------ ------ ------ ------ ------ ------ ------ ------ ------ ------
2001-03-01-09.18.23      0      0      0      0      0      0      0      0      0      0      0
2001-03-01-09.32.12      1      0      0      0      0      0      0      0      0      0      0
2001-03-01-09.47.10      2      1      0      0      3      0      0      0      0      0      0
2001-03-01-10.02.07     13      1      0      0      2      0      0      0      0      0      0
2001-03-01-10.17.05      3      0      0      0      5      0      0      0      0      0      0
2001-03-01-10.32.02     20      1      0      0      2      0      0      0      0      0      0
2001-03-01-10.47.00     19      1      0      0      0      0      0      0      0      0      0
2001-03-01-11.01.58      0      0      0      0      1      0      0      0      0      0      0
2001-03-01-11.16.51     24      2      0      0      6      0      0      0      0      0      0
2001-03-01-11.31.53      5      1      0      0      0      0      0      0      0      0      0
```

Figure 59. DB2 Statistics Stored Procedures Report

DATETIME
: Date and time that this statistics interval ended.

STORED PROCEDURES CALL
: Number of stored procedure CALL statements executed during this statistics interval.

STORED PROCEDURES ABEND
: Number of times a stored procedure terminated abnormally during this statistics interval.

STORED PROCEDURES TM-OUT
: Number of times an SQL CALL statement timed out while waiting to be scheduled during this statistics interval.

STORED PROCEDURES REJECT
: Number of times an SQL CALL statement was rejected because the procedure was in the ‘STOP ACTION(REJECT)’ state during this statistics interval.

USER FUNCTIONS EXEC
: Number of user-defined functions executed during this statistics interval (DB2 6.1 and later only).

USER FUNCTIONS ABEND
: Number of times a user-defined function abended during this statistics interval (DB2 6.1 and later only).

USER FUNCTIONS TM-OUT
: Number of times a user-defined function timed out waiting to be scheduled during this statistics interval (DB2 6.1 and later only).

USER FUNCTIONS REJECT
: Number of times a user-defined function was rejected during this statistics interval (DB2 6.1 and later only).

TRIGGERS STMACT

Number of times a statement trigger is activated during this statistics interval (DB2 6.1 and later only).

TRIGGERS ROWACT

Number of times a row trigger is activated during this statistics interval (DB2 6.1 and later only).

TRIGGERS SQLERR

Number of times an SQL error occurred during the execution of a triggered action during this statistics interval (DB2 6.1 and later only).

## STOVDFT—DB2 Statistics DDF Overview Report—Totals

The DB2 Statistics DDF Overview Report provides an overview of DDF activity. It is produced with totals (STOVDFT) or averages per remote destination, carried to one decimal place (STOVDF).

```
BMC SOFTWARE INC                             DB2 STATISTICS DDF OVERVIEW REPORT - TOTALS                               PAGE 0001
REPORT: STOVDFT                                                                                 REPORT DATE: 2001-03-25 10.25.45
DATE FROM: 2001-03-01 06.50.21                                                                                 LOCATION - SANJOSE
DATE TO:   2001-03-19 08:43:39                                                                                 SUBSYSTEM - DB2D

                        REMOTE         SQL     SQL     ROWS    ROWS    CONV    CONV   TRANS   TRANS    BLKS    BLKS    CONV
DATETIME                DESTINATION    SENT    RECV    SENT    RECV  INIT FM INIT TO  MIG FM  MIG TO   SENT    RECV   QUEUED
------------------- ---------------- ------- ------- ------- ------- ------- ------- ------- ------- ------- ------- -------
2001-03-01-06.50.21 ABCDEFGHIJKLMNOP       1     388       0       0    3168     122     221      15      10      18       3
2001-03-01-06.50.21 DEST_2                 1     388       0       0    3168     122     221      15      10      18       3
2001-03-01-06.50.21 DEST_X                 1     388       0       0    3168     122     221      15      10      18       3
2001-03-01-07.19.19 ABCDEFGHIJKLMNOP       2    3323       0       0   39067    1210   11157      55     107      66       0
2001-03-01-07.19.19 DEST_2                 2    3323       0       0   39067    1210   11157      55     107      66       0
2001-03-01-07.19.19 DEST_X                 2    3323       0       0   39067    1210   11157      55     107      66       0
```

Figure 60. DB2 Statistics DDF Overview Report - Totals

DATETIME
Date and time that this statistics interval ended.

REMOTE DESTINATION
Sixteen-character location name of the remote site.

SQL SENT
Number of SQL statements sent.

SQL RECV
Number of SQL statements received.

ROWS SENT
Number of rows sent.

ROWS RECV
Number of rows received.

CONV INIT FM
Number of conversations initiated from this site.

CONV INIT TO
Number of conversations initiated to this site.

TRANS MIG FM
Number of transactions migrated from remote locations.

TRANS MIG TO
Number of transactions migrated to remote locations.

BLKS SENT
Number of blocks sent.

BLKS RECV
Number of blocks received.

CONV QUEUED
Number of conversation requests queued.

# Lock Reports

For a statistics interval, the lock reports provide averages and totals about the amount of lock activity.

## STLOCK—DB2 Statistics Lock Report

The DB2 Statistics Lock Report includes both averages and totals.

```
BMC SOFTWARE INC                                  DB2 STATISTICS LOCK REPORT                                                PAGE -     1
REPORT: STLOCK                                                                                  REPORT DATE: 2001-03-25 15.49.33
DATE FROM: 2001-03-01 10.22.10                                                                                 LOCATION - SANJOSE
DATE TO:   2001-03-19 06.54.56                                                                                 SUBSYSTEM - DB2F

                           AVG      AVG      AVG      AVG
                    CREATE LOCK     LOCK     LATCH    OTHER                             EXCAL. ESCAL. LKTBL   DRAIN-REQ     CLAIM-REQ
DATETIME            THDS   REQ      SUSP.    SUSP.    SUSP.    DEADLOCK TIMEOUT SHRD   EXCLU. STMTS  TOTAL  FAIL   TOTAL  FAIL
------------------- ------ -------- -------- -------- -------- -------- ------- ------ ------ ------ ------ ------ ------ ------
2001-03-01-10.22.10      0      0.0      0.0      0.0      0.0        0       0      0      0      0      0      0      0      0
2001-03-01-10.51.46     13    422.5      0.0     19.4      0.0        0       0      0      0      0      1      0      0      0
2001-03-01-11.21.39      6     92.0      0.0      0.5      0.0        0       0      0      0      0      0      0      0      0
2001-03-01-11.51.33      9    613.1      0.0      0.9      0.0        0       0      0      1      0      0      0      2      0
2001-03-01-12.21.27     14     42.9      0.0      5.2      0.0        0       0      0      0      0      0      0      0      0
2001-03-01-12.51.21     11    120.0      0.0      8.5      0.0        0       0      0      0      0      0      0      0      0
2001-03-01-13.21.16      1     26.3      0.0      0.0      0.0        0       0      0      0      0      0      0      0      0
2001-03-01-13.51.10      3     21.7      0.0      0.0      0.0        0       0      0      0      0      0      0      0      0
2001-03-01-14.21.04      3     45.7      0.0      0.1      0.0        0       0      0      0      0      0      0      0      0
2001-03-01-14.50.58      6    332.8      0.0      9.4      0.0        0       0      0      0      0      0      0      0      0
2001-03-01-15.20.52      0      9.5      0.0      1.0      0.0        0       0      0      0      0      0      0      0      0
2001-03-01-15.50.46      9    389.5      0.0      8.3      0.0        0       0      0      0      0      2      1      0      0
2001-03-01-16.20.40      4     81.2      0.0     16.5      0.0        0       0      0      0      0      0      0      0      0
2001-03-01-16.50.34      7     69.8      0.0      5.8      0.0        0       0      0      0      0      0      0      0      0
2001-03-01-17.20.29      5     52.4      0.0      1.8      0.0        0       0      0      0      0      0      0      0      0
2001-03-01-17.50.23      5    245.8      0.0      0.2      0.0        0       0      0      0      0      0      0      0      0
2001-03-01-18.20.17      8    278.5      0.0      3.8      0.0        0       0      0      0      0      0      0      0      0
```

Figure 61. DB2 Statistics Lock Report

DATETIME
Date and time that this statistics interval ended.

CREATE THREADS
Number of threads created in this statistics interval.

AVG LOCK REQ
Average number of lock requests per CREATE THREAD during this statistics interval.

AVG LOCK SUSP.
Average number of times suspended due to waiting for lock per CREATE THREAD during this statistics interval.

AVG LATCH SUSP.
Average number of times suspended due to latching per CREATE THREAD during this statistics interval.

AVG OTHER SUSP.
Average number of times suspended due to other reasons per CREATE THREAD during this statistics interval.

DEADLK
Total number of deadlocks detected during this statistics interval.

TIMEOUT
Total number of lock timeouts during this statistics interval.

ESCAL. SHRD
Total number of lock escalations in shared mode during this statistics interval.

ESCAL. EXCLU
Total number of lock escalations in exclusive mode during this statistics interval.

LKTBL STMTS
Total number of SQL LOCK TABLE statements issued during this statistics interval.

DRAIN-REQUESTS TOTAL
Total number of drain requests.

DRAIN-REQUESTS FAIL
Total number of unsuccessful drain requests.

CLAIM-REQUESTS TOTAL
Total number of claim requests.

CLAIM-REQUESTS FAIL
Total number of unsuccessful claim requests.

# STGLOCK—DB2 Statistics Global Lock Report

The DB2 Statistics Global Lock Report provides information about global lock activity in a data sharing environment during each statistics interval. All values are totals.

```
BMC SOFTWARE INC                                DB2 STATISTICS GLOBAL LOCK REPORT                                  PAGE -     1
REPORT: STGLOCK                                                                                REPORT DATE: 2001-03-25 14.48.55
DATE FROM: 2001-03-01 10.36.22                                                                                LOCATION - SANJOSE
DATE TO:   2001-03-19 21.41.03                                                                                SUBSYSTEM - DB2G

                     CREATE <--------PLOCKS----------> <---------XES------------> <-------SUSPENDS--------->   INCOMP   NOTIFY
DATETIME             THREADS   LOCK    UNLOCK   CHANGE    LOCK    UNLOCK   CHANGE    IRLM      XES    FALSE   RETAINED MSG SENT
--------            -------- -------- -------- -------- -------- -------- -------- -------- -------- -------- -------- --------
2001-03-01-10.36.22        8       98       43       14      321      305        7       24        6       27        0        6
2001-03-01-10.51.19        0        2       13       23      147      160       10       10        5        0        0        0
2001-03-01-11.06.17        0        0        1        1      101      102        0        1        0        0        0        0
2001-03-01-11.21.14        0        0        0        0      101      101        0        0        0        0        0        0
2001-03-01-11.36.12        2       17        3        0      123      130        0        4       18        0        0       14
2001-03-01-11.50.49        5      119       74       21      265      291        3       59       35        1        0       51
2001-03-01-12.06.07        9      309      222       13     1172     1129       20       45       12       11        0       75
2001-03-01-12.21.05       10      161      119        5      867      831        5        6        4       23        0       74
2001-03-01-12.34.43       11      128       83       52      715      751        1      112       27        9        0       18
2001-03-01-12.51.00       13      426      296       25     1238     1206       21       71       27       73        0      126
```

Figure 62.  DB2 Statistics Global Lock Report

A physical lock (P-lock) is a lock used only by data sharing and is acquired by DB2 to provide consistency on data cached in different DB2 subsystems. P-locks are owned by the subsystem, not by the transaction. XES is the cross-system extended services component of MVS.

DATETIME
: Date and time that this statistics interval ended.

CREATE THREADS
: Number of threads created in this statistics interval.

PLOCKS-LOCK
: Number of lock requests for P-locks during this statistics interval.

PLOCKS-UNLOCK
: Number of unlock requests for P-locks during this statistics interval.

PLOCKS-CHANGE
: Number of change requests for P-locks during this statistics interval.

XES-LOCK
: Total number of lock requests propagated to MVS XES synchronously under the user’s execution unit during this statistics interval. This includes both L-Locks and P-locks. It is not incremented if the request is suspended before going to XES.

XES-UNLOCK
: Total number of change requests propagated to MVS XES synchronously under the user’s execution unit during this statistics interval. This includes both L-Locks and P-locks. It is not incremented if the request is suspended before going to XES.

XES-CHANGE
: Total number of resources propagated to MVS XES synchronously under the user’s execution unit for unlock requests during this statistics interval. This includes both L-Locks and P-locks. It is not incremented if the request is suspended before going to XES. It can be incremented each time a global lock is released by the generic unlock.

SUSPENDS-IRLM
Total number of suspends caused by IRLM global resource contention (IRLM lock states were in conflict on the same resource) during this statistics interval. Global contention requires inter-system communication to resolve the lock conflict. This value plus the two other suspends gives the total number of suspends caused by global contention.

SUSPENDS-XES
Total number of suspends caused by MVS XES global resource contention (MVS XES lock states were in conflict but IRLM lock states were not) during this statistics interval. IRLM has many lock states, but XES is aware only of the exclusive and shared lock states.

SUSPENDS-FALSE
Total number of suspends because of false contentions during this statistics interval. This occurs when different resource names hash to the same entry in the coupling facility lock table. This causes MVS XES to detect contention on the hash class. When MVS XES determines that there is no real conflict on the resource, the contention is called *false*.

INCOMP RETAINED
Total number of global lock or change requests denied because of an incompatible retained lock during this statistics interval.

NOTIFY MESSAGE SENT
Total number of notify messages sent during this statistics interval.

# EDM Pool Reports

For a statistics interval, the EDM pool reports provide information about the amount of Environmental Descriptor Manager pool activity.

## STEDM—DB2 Statistics EDM Pool Report

The DB2 Statistics EDM Pool Report displays the status and activity of the Environmental Descriptor Manager pool. It includes both percentages and totals.

```
BMC SOFTWARE INC                                  DB2 STATISTICS EDM POOL REPORT                                  PAGE -     1
REPORT: STEDM                                                                                     REPORT DATE: 2000-02-25 10.13.42
DATE FROM: 1999-03-04  00.18.46                                                                           LOCATION - DB2J
DATE TO:   1999-03-04  20.52.26                                                                           SUBSYSTEM - DB1J

                    CREATE <----EDM TOTALS---->  DBD    CT   PKGTBL  SKCT   SKPT   DBD    DBD    CT     CT    PKG    PKG
DATETIME            THDS    PAGES  FREE   FAILS  PAGES  PAGES  PAGES  PAGES  PAGES  REQS   LOADS  REQS   LOADS  REQS   LOADS
------------------- ------ ------ ------ ------ ------ ------ ------ ------ ------ ------ ------ ------ ------ ------ ------
1999-03-04-00.18.46      0   7343   7293      0     48      0      0      1      1      0      0      0      0      0      0
1999-03-04-00.48.40      0   7343   7293      0     48      0      0      1      1      0      0      0      0      0      0
1999-03-04-01.18.33      0   7343   7293      0     48      0      0      1      1      0      0      0      0      0      0
1999-03-04-10.11.30      0   7343   7321      0     22      0      0      0      0      5      2      0      0      0      0
1999-03-04-10.41.50     10   7343   7321      0     22      0      0      0      0      8      0      0      0      0      0
1999-03-04-11.11.43      0   7343   7321      0     22      0      0      0      0      0      0      0      0      0      0
1999-03-04-11.41.36      0   7343   7321      0     22      0      0      0      0      0      0      0      0      0      0
1999-03-04-12.11.29      1   7343   7321      0     22      0      0      0      0      6      0      0      0      0      0
1999-03-04-12.41.22      0   7343   7321      0     22      0      0      0      0      0      0      0      0      0      0
1999-03-04-13.11.15      0   7343   7321      0     22      0      0      0      0      0      0      0      0      0      0
1999-03-04-13.41.08      0   7343   7321      0     22      0      0      0      0      0      0      0      0      0      0
1999-03-04-14.11.02      1   7343   7321      0     22      0      0      0      0      4      0      0      0      0      0
1999-03-04-14.40.55      1   7343   7321      0     22      0      0      0      0      7      0      0      0      0      0
1999-03-04-15.10.48      0   7343   7321      0     22      0      0      0      0      0      0      0      0      0      0
1999-03-04-15.40.41      0   7343   7321      0     22      0      0      0      0      0      0      0      0      0      0
1999-03-04-16.09.34      0   7343   7321      0     22      0      0      0      0      0      0      0      0      0      0
1999-03-04-16.39.27      0   7343   7321      0     22      0      0      0      0      0      0      0      0      0      0
1999-03-04-17.09.21      0   7343   7321      0     22      0      0      0      0      0      0      0      0      0      0
1999-03-04-17.39.14      0   7343   7321      0     22      0      0      0      0      0      0      0      0      0      0
1999-03-04-18.09.07      1   7343   7321      0     22      0      0      0      0      2      0      0      0      0      0
1999-03-04-18.39.00      0   7343   7321      0     22      0      0      0      0      0      0      0      0      0      0
1999-03-04-19.08.53      1   7343   7321      0     22      0      0      0      0      4      0      0      0      0      0
1999-03-04-19.38.46      1   7343   7321      0     22      0      0      0      0      2      0      0      0      0      0
1999-03-04-20.08.39      0   7343   7321      0     22      0      0      0      0      0      0      0      0      0      0
1999-03-04-20.38.33      0   7343   7321      0     22      0      0      0      0      0      0      0      0      0      0
1999-03-04-20.52.26     10   7343   7321      0     22      0      0      0      0      9      0      0      0      0      0
```

Figure 63.  DB2 Statistics EDM Pool Report

DATETIME
: Date and time that this statistics interval ended.

CREATE THREADS
: Number of threads created in this statistics interval.

EDM TOTALS PAGES
: Total number of pages in the EDM pool during this statistics interval.

EDM TOTALS FREE
: Total number of EDM pool pages available during this statistics interval.

EDM TOTALS FAILS
: Total number of requests to DB2 that failed because of EDM out-of-space conditions during this statistics interval.

DBD PAGES
: Total number of pages in the EDM pool used for database descriptors during this statistics interval.

CT PAGES
: Total number of pages in the EDM pool used for cursor table sections during this statistics interval.

PKGTBL PAGES
: Total number of pages in the EDM pool used for the package table.

SKCT PAGES
: Total number of pages in the EDM pool used for skeleton cursor tables during this statistics interval.

SKPT PAGES
: Total number of pages in the EDM pool used for skeleton package tables during this statistics interval.

DBD REQS
: Total number of requests for database descriptors during this statistics interval.

DBD LOADS
: Total number of database descriptors loaded from the DB2 directory during this statistics interval.

CT REQS
: Total number of requests for cursor table sections during this statistics interval.

CT LOADS
: Total number of cursor table sections loaded from the DB2 directory during this statistics interval.

PKG REQS
: Total number of requests for package table sections during this statistics interval.

PKG LOADS
: Total number of load package table sections from DASD.

## STEDM2—DB2 Statistics EDM Pool Extended Report

The DB2 Statistics EDM Pool Extended Report displays the status and activity of the Environmental Descriptor Manager pool. It includes both percentages and totals.

```
BMC SOFTWARE INC                                    DB2 STATISTICS EDM POOL EXTENDED REPORT                          PAGE -     1
REPORT: STEDM2                                                                                          REPORT DATE: 2000-02-25 10.13.42
DATE FROM: 1999-03-04  00.18.46                                                                              LOCATION - DB2J
DATE TO:   1999-03-04  20.52.26                                                                             SUBSYSTEM - DB1J

                    CREATE <DYNAMIC STMT CACHE> <----DATA SPACE---->
DATETIME            THDS    PAGES  REQS  INSERT  PAGES  FREE   FAILS
------------------- ------ ------ ------ ------ ------ ------ ------
1999-03-04-00.18.46      0      3      0      0  10240  10237      0
1999-03-04-00.48.40      0      3      0      0  10240  10237      0
1999-03-04-01.18.33      0      3      0      0  10240  10237      0
1999-03-04-10.11.30      0      0      0      0  10240  10240      0
1999-03-04-10.41.50     10      0      0      0  10240  10240      0
1999-03-04-11.11.43      0      0      0      0  10240  10240      0
1999-03-04-11.41.36      0      0      0      0  10240  10240      0
1999-03-04-12.11.29      1      0      0      0  10240  10240      0
1999-03-04-12.41.22      0      0      0      0  10240  10240      0
1999-03-04-13.11.15      0      0      0      0  10240  10240      0
1999-03-04-13.41.08      0      0      0      0  10240  10240      0
1999-03-04-14.11.02      1      0      0      0  10240  10240      0
1999-03-04-14.40.55      1      0      0      0  10240  10240      0
1999-03-04-15.10.48      0      0      0      0  10240  10240      0
1999-03-04-15.40.41      0      0      0      0  10240  10240      0
1999-03-04-16.09.34      0      0      0      0  10240  10240      0
1999-03-04-16.39.27      0      0      0      0  10240  10240      0
1999-03-04-17.09.21      0      0      0      0  10240  10240      0
1999-03-04-17.39.14      0      0      0      0  10240  10240      0
1999-03-04-18.09.07      1      0      0      0  10240  10240      0
1999-03-04-18.39.00      0      0      0      0  10240  10240      0
1999-03-04-19.08.53      1      0      0      0  10240  10240      0
1999-03-04-19.38.46      1      0      0      0  10240  10240      0
1999-03-04-20.08.39      0      0      0      0  10240  10240      0
1999-03-04-20.38.33      0      0      0      0  10240  10240      0
1999-03-04-20.52.26     10      0      0      0  10240  10240      0
```

Figure 64. DB2 Statistics EDM Pool Extended Report

DATETIME
Date and time that this statistics interval ended.

CREATE THREADS
Number of threads created in this statistics interval.

DYNAMIC STMT CACHE PAGES
Total number of pages in the EDM pool used for the dynamic statement cache (DB2 5.1 and later only).

DYNAMIC STMT CACHE REQS
Number of requests to the dynamic statement cache (DB2 5.1 and later only).

DYNAMIC STMT CACHE INSERT
Number of inserts into the dynamic statement cache (DB2 5.1 and later only).

DATA SPACE PAGES
Number of pages in the data space used by the EDM pool (DB2 6.1 and later only).

DATA SPACE FREE
Number of free pages in the data space free chain (DB2 6.1 and later only).

DATA SPACE FAILS
Number of failures because the data space is full (DB2 6.1 and later only).

## Buffer Pool Reports

For a statistics interval, the buffer pool reports provide information about the activity (read and write, prefetch, hiperpool, exceptions, RID) of all DB2 buffer pools.

## STBFpVP—DB2 Statistics Buffer Pool Activity Report

The DB2 Statistics Buffer Pool Activity Report provides information about the activity of all DB2 buffer pools, which are grouped by

| | |
|---|---|
| ALL | STBFAVP |
| x | STBFXVP |

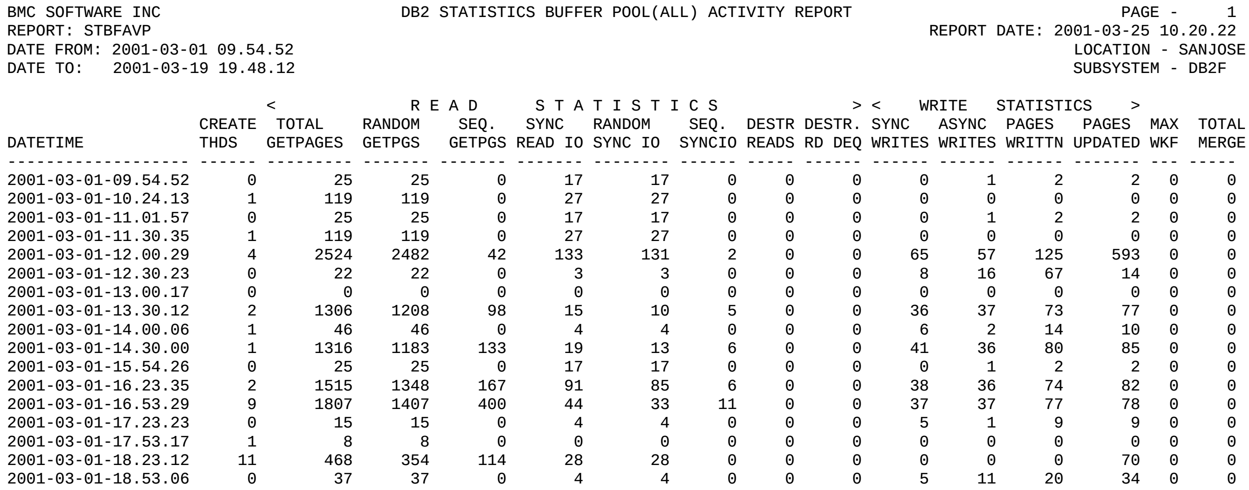
By default, STBFXVP reports about buffer pool BP0. For a different, single buffer pool, customize a copy of BBPARM member STBFXVP as follows:

- WHERE clause of SQL statement

  Change `BPNAME = 'BP0 '` to `BPNAME = BPx '`, where `x` is the buffer pool ID of the single buffer pool to be reported (BP0-BP49, BP8K-BP8K9, BP16K-BP16K9, BP32K-BP32K9).

- Title line

  Change (`BP0`) to (`BPx`), where `x` is the buffer pool ID of the single buffer pool to be reported.



```
BMC SOFTWARE INC                         DB2 STATISTICS BUFFER POOL(ALL) ACTIVITY REPORT                            PAGE -     1
REPORT: STBFAVP                                                                                 REPORT DATE: 2001-03-25 10.20.22
DATE FROM: 2001-03-01 09.54.52                                                                                 LOCATION - SANJOSE
DATE TO:   2001-03-19 19.48.12                                                                                 SUBSYSTEM - DB2F

                           <              R E A D      S T A T I S T I C S              > <   WRITE   STATISTICS    >
                    CREATE  TOTAL    RANDOM    SEQ.   SYNC   RANDOM    SEQ.  DESTR DESTR. SYNC   ASYNC  PAGES   PAGES  MAX  TOTAL
DATETIME            THDS   GETPAGES  GETPGS   GETPGS READ IO SYNC IO  SYNCIO READS RD DEQ WRITES WRITES WRITTN UPDATED WKF  MERGE
------------------- ------ --------- ------- ------- ------- -------- ------ ----- ------ ------ ------ ------ ------- ---  -----
2001-03-01-09.54.52      0        25      25       0      17       17      0     0      0      0      1      2       2  0     0
2001-03-01-10.24.13      1       119     119       0      27       27      0     0      0      0      0      0       0  0     0
2001-03-01-11.01.57      0        25      25       0      17       17      0     0      0      0      1      2       2  0     0
2001-03-01-11.30.35      1       119     119       0      27       27      0     0      0      0      0      0       0  0     0
2001-03-01-12.00.29      4      2524    2482      42     133      131      2     0      0     65     57    125     593  0     0
2001-03-01-12.30.23      0        22      22       0       3        3      0     0      0      8     16     67      14  0     0
2001-03-01-13.00.17      0         0       0       0       0        0      0     0      0      0      0      0       0  0     0
2001-03-01-13.30.12      2      1306    1208      98      15       10      5     0      0     36     37     73      77  0     0
2001-03-01-14.00.06      1        46      46       0       4        4      0     0      0      6      2     14      10  0     0
2001-03-01-14.30.00      1      1316    1183     133      19       13      6     0      0     41     36     80      85  0     0
2001-03-01-15.54.26      0        25      25       0      17       17      0     0      0      0      1      2       2  0     0
2001-03-01-16.23.35      2      1515    1348     167      91       85      6     0      0     38     36     74      82  0     0
2001-03-01-16.53.29      9      1807    1407     400      44       33     11     0      0     37     37     77      78  0     0
2001-03-01-17.23.23      0        15      15       0       4        4      0     0      0      5      1      9       9  0     0
2001-03-01-17.53.17      1         8       8       0       0        0      0     0      0      0      0      0       0  0     0
2001-03-01-18.23.12     11       468     354     114      28       28      0     0      0      0      0      0      70  0     0
2001-03-01-18.53.06      0        37      37       0       4        4      0     0      0      5     11     20      34  0     0
```

Figure 65.  DB2 Statistics Buffer Pool Activity Report

DATETIME
: Date and time that this statistics interval ended.

CREATE THDS
: Number of threads created in this statistics interval.

**Note:** This is not applicable for the STBFXVP report.

READ STATISTICS TOTAL GETPAGES
: Number of GETPAGE requests during this statistics interval. It includes both conditional and non-conditional, successful and unsuccessful requests.

READ STATISTICS RANDOM GETPGS
: The difference between total GETPAGEs and sequential GETPAGEs during this statistics interval.

READ STATISTICS SEQ. GETPGS
: Number of GETPAGEs issued by sequential requests during this statistics interval.

READ STATISTICS SYNC READ I/O
: Total number of synchronous read I/O operations performed during this statistics interval.

READ STATISTICS RANDOM SYNC IO
: The difference between total read I/O and sequential read I/O during this statistics interval.

READ STATISTICS SEQ. SYNCIO
: Number of synchronous read I/O operations performed by sequential requests during this statistics interval.

READ STATISTICS DESTR READS
: Number of pages for which a destructive read was requested during this statistics interval.

READ STATISTICS DESTR. RD DEQ
: Number of pages removed from the data set deferred write queue for destructive read requests during this statistics interval.

WRITE STATISTICS SYNC WRITES
: Number of immediate (synchronous) write I/Os during this statistics interval.

WRITE STATISTICS ASYNC WRITES
: Number of asynchronous write I/O operations performed during this statistics interval.

WRITE STATISTICS PAGES WRITTN
: Number of pages in the buffer pool written to DASD during this statistics interval.

WRITE STATISTICS PAGES UPDATED
: Number of times a page update occurs against a page during this statistics interval.

MAX WKFILE
: Maximum number of work files ever concurrently used during merge processing during this statistics interval.

TOTAL MERGES
: Total number of merge passes for DB2 sort activities during this statistics interval.

## STBFpPF—DB2 Statistics Buffer Pool Prefetch Activity Report

The DB2 Statistics Buffer Pool Prefetch Activity Report provides information about the prefetch activity of all DB2 buffer pools, which are grouped by

| | |
|---|---|
| ALL | STBFAPF |
| x | STBFXPF |

By default, STBFXPF reports about buffer pool BP0. For a different, single buffer pool, customize a copy of BBPARM member STBFXPF as follows:

- WHERE clause of SQL statement

  Change `BPNAME = 'BP0 '` to `BPNAME = BPx  '`, where `x` is the buffer pool ID of the single buffer pool to be reported (BP0-BP49, BP8K-BP8K9, BP16K-BP16K9, BP32K-BP32K9).

- Title line

  Change (`BP0`) to (`BPx`), where `x` is the buffer pool ID of the single buffer pool to be reported.

```
BMC SOFTWARE INC                   DB2 STATISTICS BUFFER POOL(ALL) PREFETCH ACTIVITY REPORT                         PAGE -     2
REPORT: STBFAPF                                                                                     REPORT DATE: 2001-03-25 14.48.55
DATE FROM: 2001-03-01 10.36.22                                                                                    LOCATION - SANJOSE
DATE TO:   2001-03-19 21.41.03                                                                                    SUBSYSTEM - DB2G

                    CREATE  <  PREFETCH REQUESTS  > <   ASYNC READ I/O   >   <  TOTAL PAGES READ   >    SYNC    NO      NO
DATETIME            THDS    SEQ.    LIST    DYNAMIC   SEQ.   LIST  DYNAMIC   SEQ.    LIST    DYNAMIC   READS   BUFFER  READ ENG
------------------- ------  ------- ------- ------- ------- ------ -------   ------- ------- -------   ------- ------- --------
2001-03-01-11.21.14      0        0       0       0       0      0       0         0       0       0         0       0        0
2001-03-01-11.36.12      2        0       0       0       0      0       0         0       0       0         0       0        0
2001-03-01-11.50.49      5        0       0       0       0      0       0         0       0       0         0       0        0
2001-03-01-12.06.07      9      584       0       0     402      0       0      3098       0       0       563       0        0
2001-03-01-12.21.05     10      149       0       0     105      0       0       809       0       0       143       0        0
2001-03-01-12.34.43     11        0       0       0       0      0       0         0       0       0         2       0        0
2001-03-01-12.51.00     13      441       0       0     300      0       0      2335       0       0       353       0        0
2001-03-01-13.05.58      0        0       0       0       0      0       0         0       0       0         0       0        0
2001-03-01-13.20.55      0        0       0       0       0      0       0         0       0       0         0       0        0
2001-03-01-13.35.53      0        0       0       0       0      0       0         0       0       0         0       0        0
```

Figure 66. DB2 Statistics Buffer Pool Prefetch Activity Report

DATETIME
: Date and time that this statistics interval ended.

CREATE THDS
: Number of threads created in this statistics interval.

  **Note:** This is not applicable for the STBFXPF report.

PREFETCH REQUESTS SEQ.
: Number of sequential prefetch requests during this statistics interval.

PREFETCH REQUESTS LIST
: Number of list prefetch requests during this statistics interval.

PREFETCH REQUESTS DYNAMIC
: Number of dynamic prefetch requests during this statistics interval.

ASYNC READ I/O SEQ.
: Number of sequential prefetch (asynchronous) read I/Os during this statistics interval.

ASYNC READ I/O LIST
Number of list prefetch (asynchronous) read I/Os during this statistics interval.

ASYNC READ I/O DYNAMIC
Number of dynamic prefetch (asynchronous) read I/Os during this statistics interval.

TOTAL PAGES READ SEQ.
Total number of pages read due to sequential prefetch during this statistics interval.

TOTAL PAGES READ LIST
Total number of pages read due to list prefetch during this statistics interval.

TOTAL PAGES READ DYNAMIC
Total number of pages read due to dynamic prefetch during this statistics interval.

SYNC READS
Number of synchronous read I/Os performed for sequential requests during this statistics interval.

NO BUFFER
Number of times sequential prefetch was disabled because buffers were not available during this statistics interval.

NO READ ENG
Number of times sequential prefetch was disabled due to an unavailable read engine during this statistics interval.

## STBFpHP—DB2 Statistics Hiperpool Activity Report

The DB2 Statistics Hiperpool Activity Report provides information about the hiperpool activity of all DB2 buffer pools, which are grouped by

| | |
|---|---|
| ALL | STBFAHP |
| x | STBFXHP |

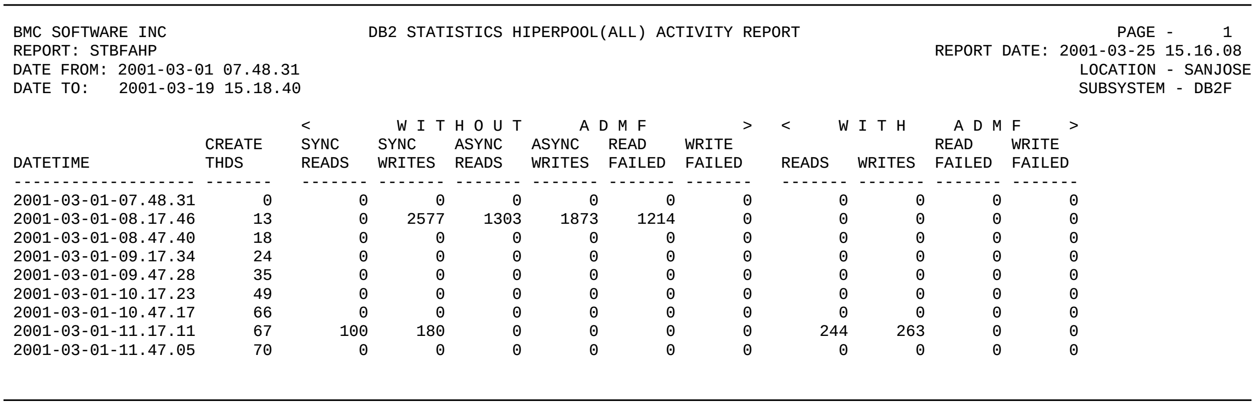
By default, STBFXHP reports about buffer pool BP0. For a different, single buffer pool, customize a copy of BBPARM member STBFXHP as follows:

- WHERE clause of SQL statement

  Change `BPNAME = 'BP0 '` to `BPNAME = BPx  '`, where `x` is the buffer pool ID of the single buffer pool to be reported (BP0-BP49, BP8K-BP8K9, BP16K-BP16K9, BP32K-BP32K9).

- Title line

  Change (`BP0`) to (`BPx`), where `x` is the buffer pool ID of the single buffer pool to be reported.



```
BMC SOFTWARE INC                     DB2 STATISTICS HIPERPOOL(ALL) ACTIVITY REPORT                                 PAGE -     1
REPORT: STBFAHP                                                                                 REPORT DATE: 2001-03-25 15.16.08
DATE FROM: 2001-03-01 07.48.31                                                                                 LOCATION - SANJOSE
DATE TO:   2001-03-19 15.18.40                                                                                 SUBSYSTEM - DB2F

                              <         W I T H O U T      A D M F          >   <     W I T H     A D M F     >
                    CREATE    SYNC    SYNC    ASYNC   ASYNC   READ    WRITE                     READ    WRITE
DATETIME            THDS      READS   WRITES  READS   WRITES  FAILED  FAILED    READS   WRITES  FAILED  FAILED
------------------- -------   ------- ------- ------- ------- ------- -------   ------- ------- ------- -------
2001-03-01-07.48.31       0         0       0       0       0       0       0         0       0       0       0
2001-03-01-08.17.46      13         0    2577    1303    1873    1214       0         0       0       0       0
2001-03-01-08.47.40      18         0       0       0       0       0       0         0       0       0       0
2001-03-01-09.17.34      24         0       0       0       0       0       0         0       0       0       0
2001-03-01-09.47.28      35         0       0       0       0       0       0         0       0       0       0
2001-03-01-10.17.23      49         0       0       0       0       0       0         0       0       0       0
2001-03-01-10.47.17      66         0       0       0       0       0       0         0       0       0       0
2001-03-01-11.17.11      67       100     180       0       0       0       0       244     263       0       0
2001-03-01-11.47.05      70         0       0       0       0       0       0         0       0       0       0
```

Figure 67. DB2 Statistics Hiperpool Activity Report

DATETIME
Date and time that this statistics interval ended.

CREATE THDS
Number of threads created in this statistics interval.

**Note:** This is not applicable for the STBFXHP report.

WITHOUT ADMF SYNC READS
Number of pages read from a hiperpool to a virtual buffer pool synchronously using MVPG during this statistics interval.

WITHOUT ADMF SYNC WRITES
Number of pages written from a virtual buffer pool to a hiperpool synchronously using MVPG or CWRITE during this statistics interval.

WITHOUT ADMF ASYNC READS
Number of pages read from a hiperpool to a virtual buffer pool asynchronously using MVPG during this statistics interval.

WITHOUT ADMF ASYNC WRITES
Number of pages written from a virtual buffer pool to a hiperpool asynchronously using MVPG or CWRITE during this statistics interval.

WITHOUT ADMF READ FAILED
Number of unsuccessful hiperpool to virtual buffer pool MVPG reads during this statistics interval.

WITHOUT ADMF WRITE FAILED
Number of unsuccessful virtual buffer pool to hiperpool page writes using MVPG or CWRITE during this statistics interval.

WITH ADMF READS
Number of pages read from a hiperpool to a virtual buffer pool asynchronously using the asynchronous data mover facility during this statistics interval.

WITH ADMF WRITES
Number of pages written from a virtual buffer pool to a hiperpool asynchronously using the asynchronous data mover facility during this statistics interval.

WITH ADMF READ FAILED
Number of unsuccessful hiperpool to virtual buffer pool reads using the asynchronous data mover facility during this statistics interval.

WITH ADMF WRITE FAILED
Number of unsuccessful virtual buffer pool to hiperpool writes using the asynchronous data mover facility during this statistics interval.

## STBFpEX—DB2 Statistics Buffer Pool Exceptions Report

The DB2 Statistics Buffer Pool Exceptions Report provides information about the exceptions activity of all DB2 buffer pools, which are grouped by

| | |
|---|---|
| ALL | STBFAEX |
| x | STBFXEX |

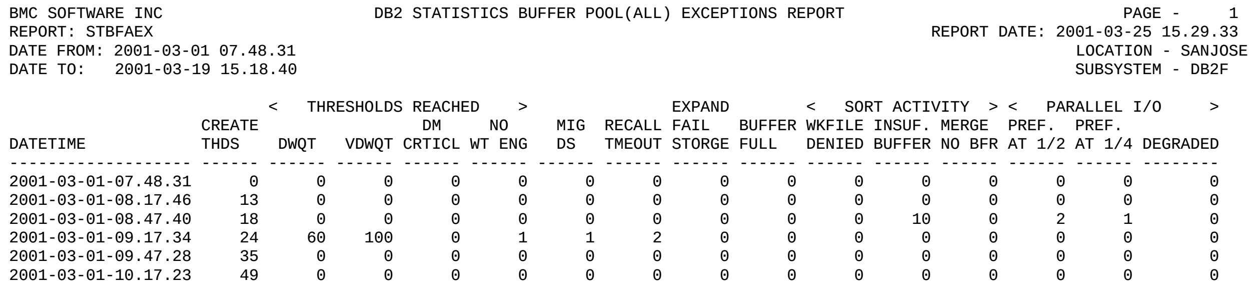
By default, STBFXEX reports about buffer pool BP0. For a different, single buffer pool, customize a copy of BBPARM member STBFXEX as follows:

- WHERE clause of SQL statement

  Change `BPNAME = 'BP0 '` to `BPNAME = BPx  '`, where `x` is the buffer pool ID of the single buffer pool to be reported (BP0-BP49, BP8K-BP8K9, BP16K-BP16K9, BP32K-BP32K9).

- Title line

  Change (`BP0`) to (`BPx`), where `x` is the buffer pool ID of the single buffer pool to be reported.



```
BMC SOFTWARE INC                      DB2 STATISTICS BUFFER POOL(ALL) EXCEPTIONS REPORT                             PAGE -     1
REPORT: STBFAEX                                                                                 REPORT DATE: 2001-03-25 15.29.33
DATE FROM: 2001-03-01 07.48.31                                                                                 LOCATION - SANJOSE
DATE TO:   2001-03-19 15.18.40                                                                                 SUBSYSTEM - DB2F

                           <   THRESHOLDS REACHED    >               EXPAND        <   SORT ACTIVITY  > <   PARALLEL I/O     >
                    CREATE                 DM     NO     MIG    RECALL FAIL   BUFFER WKFILE INSUF. MERGE  PREF.  PREF.
DATETIME            THDS    DWQT   VDWQT CRTICL WT ENG   DS   TMEOUT STORGE FULL   DENIED BUFFER NO BFR AT 1/2 AT 1/4 DEGRADED
------------------- ------ ------ ------ ------ ------ ------ ------ ------ ------ ------ ------ ------ ------ ------ --------
2001-03-01-07.48.31      0      0      0      0      0      0      0      0      0      0      0      0      0      0        0
2001-03-01-08.17.46     13      0      0      0      0      0      0      0      0      0      0      0      0      0        0
2001-03-01-08.47.40     18      0      0      0      0      0      0      0      0      0     10      0      2      1        0
2001-03-01-09.17.34     24     60    100      0      1      1      2      0      0      0      0      0      0      0        0
2001-03-01-09.47.28     35      0      0      0      0      0      0      0      0      0      0      0      0      0        0
2001-03-01-10.17.23     49      0      0      0      0      0      0      0      0      0      0      0      0      0        0
```

Figure 68. DB2 Statistics Buffer Pool Exceptions Report

DATETIME
Date and time that this statistics interval ended.

CREATE THDS
Number of threads created in this statistics interval.

**Note:** This is not applicable for the STBFXEX report.

THRESHOLDS REACHED DWQT
Number of times the Deferred Write threshold was reached during this statistics interval.

THRESHOLDS REACHED VDWQT
Number of times the Vertical Deferred Write threshold was reached during this statistics interval.

THRESHOLDS REACHED DM CRITICAL
Number of times the Data Manager Buffer Critical threshold was reached during this statistics interval.

THRESHOLDS REACHED NO WT ENG

Number of times a write engine was not available for asynchronous write I/O during this statistics interval.

MIG DS

Number of times migrated data sets were encountered during this statistics interval.

RECALL TIMEOUT

Number of recall timeouts during this statistics interval.

EXPAND FAIL STORAGE

Number of expansion failures due to a shortage of virtual storage during this statistics interval.

BUFFER FULL

Number of times a usable buffer could not be located in the virtual buffer pool because the virtual buffer pool was full during this statistics interval.

SORT ACTIVITY WKFILE DENIED

Total number of work files rejected during all merge passes because of insufficient buffer resources during this statistics interval.

SORT ACTIVITY INSUF. BUFFER

Number of work files not created because of insufficient buffer resources during this statistics interval.

SORT ACTIVITY MERGE NO BFR

Number of times the merge pass was not efficiently performed because of a shortage of buffer space during this statistics interval.

PARALLEL I/O PREF. AT 1/2

Total number of occurrences when the prefetch quantity decreased from normal to half of normal during this statistics interval. Normal is based on the buffer pool size.

PARALLEL I/O PREF. AT 1/4

Total number of occurrences when the prefetch quantity decreased from half of normal to one-quarter of normal during this statistics interval. Normal is based on the buffer pool size.

PARALLEL I/O DEGRADED

Number of times DB2 could not allocate the requested number of buffer pages to allow a parallel group to run to the planned degree during this statistics interval. This is caused by a storage shortage in the buffer pool.

# STRID—DB2 Statistics RID Activity Report

The DB2 Statistics RID Activity Report provides an overview of RID (Record ID) pool usage.

```
BMC SOFTWARE INC                             DB2 STATISTICS RID ACTIVITY REPORT                                 PAGE    1
REPORT: STRID                                                                                   REPORT DATE: 2001-03-25 10.25.45
DATE FROM: 2001-03-01  07.50.11                                                                                LOCATION - SANJOSE
DATE TO:   2001-03-19  08:43:39                                                                                SUBSYSTEM - DB2D

                                               INSERT/    RID      RID      RID      RID      RID      RID
                      NUM                      UPDATE/    CURR     MAX      OVER     OVER     NO       MAX
DATETIME              THRDS    SELECTS  DELETE    BLOCKS   BLOCKS     RDS      DM    STORAGE  CONCURR
------------------- -------- -------- -------- -------- -------- -------- -------- -------- --------
2001-03-01-07.50.11        1      206        0      190       10       12       12        3        0
2001-03-01-08.20.05        1      411        0      221       17        8       18        2        0
2001-03-01-08.50.01        1      388        0      377       15       10       22        3        0
2001-03-01-09.19.55        2     3323        2     9987       55      107       66        0        0
2001-03-01-09.49.53        2     4019        0      650       45       98       43        1        0
2001-03-01-10.19.45        2     2503        0    11157       63      135       80        0        0
```

Figure 69. DB2 Statistics RID Activity Report

DATETIME
Date and time that this statistics interval ended.

NUM THRDS
Number of threads created in this statistics interval.

SELECTS
Number of SELECT statements processed during this statistics interval.

INSERT/UPDATE/DELETE
Number of INSERT, UPDATE, and DELETE statements processed during this statistics interval.

RID CURR BLOCKS
Current number of 4K blocks used for the RID lists during this statistics interval.

RID MAX BLOCKS
Maximum number of 4K blocks used for the RID lists during this statistics interval.

RID OVER RDS
Number of times the number of RIDs for a request exceeded the RDS (Relational Data System) limit during this statistics interval.

RID OVER DM
Number of times the number of RIDs for a request exceeded the DM (Data Manager) limit during this statistics interval.

RID NO STORAGE
Number of times the pool size was insufficient to meet the number of concurrent RID lists during this statistics interval.

RID MAX CONCURR
Number of times the maximum number of concurrent RID lists was exceeded during this statistics interval.

# STGBp—DB2 Statistics Global BPool Activity Report

The DB2 Statistics Global BPool Activity Report provides information about the activity of all DB2 global buffer pools, which are grouped by

| | |
|---|---|
| ALL | STGBA |
| x | STGBX |

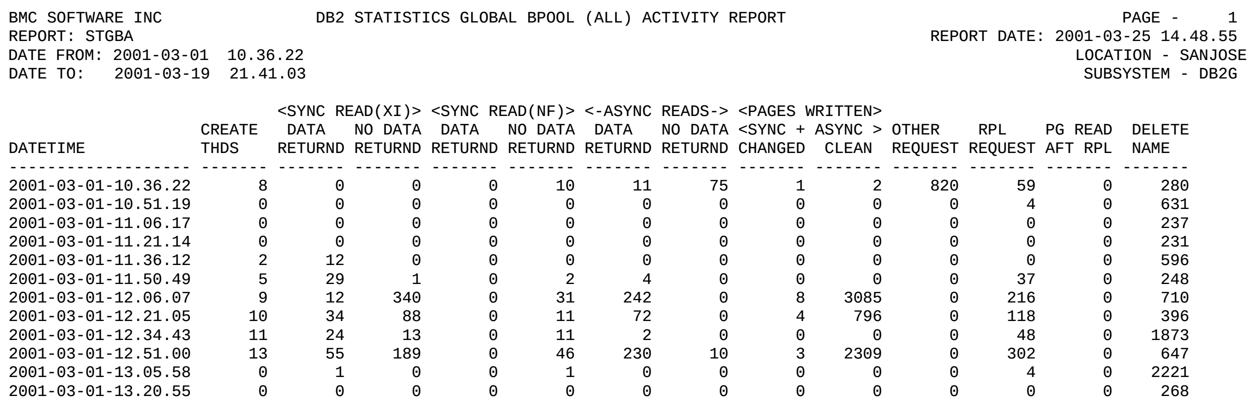
By default, STGBX reports about global buffer pool GBP0. For a different, single buffer pool, customize a copy of BBPARM member STGBX as follows:

- WHERE clause of SQL statement

  Change BPNAME = 'BP0 ' to BPNAME = BPx  ', where x is the buffer pool ID of the single buffer pool to be reported (BP0-BP49, BP8K-BP8K9, BP16K-BP16K9, BP32K-BP32K9).

- Title line

  Change (BP0) to (BPx), where x is the buffer pool ID of the single buffer pool to be reported.



```
BMC SOFTWARE INC                DB2 STATISTICS GLOBAL BPOOL (ALL) ACTIVITY REPORT                                   PAGE -     1
REPORT: STGBA                                                                                   REPORT DATE: 2001-03-25 14.48.55
DATE FROM: 2001-03-01  10.36.22                                                                                LOCATION - SANJOSE
DATE TO:   2001-03-19  21.41.03                                                                                 SUBSYSTEM - DB2G

                            <SYNC READ(XI)> <SYNC READ(NF)> <-ASYNC READS-> <PAGES WRITTEN>
                    CREATE   DATA   NO DATA  DATA   NO DATA  DATA   NO DATA <SYNC + ASYNC > OTHER    RPL    PG READ  DELETE
DATETIME            THDS    RETURND RETURND RETURND RETURND RETURND RETURND CHANGED   CLEAN REQUEST REQUEST AFT RPL  NAME
------------------- ------- ------- ------- ------- ------- ------- ------- ------- ------- ------- ------- ------- -------
2001-03-01-10.36.22       8       0       0       0      10      11      75       1       2     820      59       0     280
2001-03-01-10.51.19       0       0       0       0       0       0       0       0       0       0       4       0     631
2001-03-01-11.06.17       0       0       0       0       0       0       0       0       0       0       0       0     237
2001-03-01-11.21.14       0       0       0       0       0       0       0       0       0       0       0       0     231
2001-03-01-11.36.12       2      12       0       0       0       0       0       0       0       0       0       0     596
2001-03-01-11.50.49       5      29       1       0       2       4       0       0       0       0      37       0     248
2001-03-01-12.06.07       9      12     340       0      31     242       0       8    3085       0     216       0     710
2001-03-01-12.21.05      10      34      88       0      11      72       0       4     796       0     118       0     396
2001-03-01-12.34.43      11      24      13       0      11       2       0       0       0       0      48       0    1873
2001-03-01-12.51.00      13      55     189       0      46     230      10       3    2309       0     302       0     647
2001-03-01-13.05.58       0       1       0       0       1       0       0       0       0       0       4       0    2221
2001-03-01-13.20.55       0       0       0       0       0       0       0       0       0       0       0       0     268
```

Figure 70. DB2 Statistics Global BPool Activity Report

DATETIME

Date and time that this statistics interval ended.

CREATE THDS

Number of threads created in this statistics interval.

**Note:** This is not applicable for the STGBX report.

SYNC READ(XI) - DATA RETURND

Total number of coupling facility read requests required because the buffer was marked invalid in the local pool during this statistics interval. Data was returned from the group buffer pool.

SYNC READ(XI) - NO DATA RETURND

Total number of coupling facility read requests required because the buffer was marked invalid in the local pool during this statistics interval. Data was not returned from the group buffer pool and had to be read from DASD.

SYNC READ(NF) - DATA RETURND

Total number of coupling facility read requests required because the requested page was not found in the buffer pool during this statistics interval. Data was returned from the coupling facility.

SYNC READ(NF) - NO DATA RETURND

Total number of coupling facility read requests required because the requested page was not found in the buffer pool during this statistics interval. Data was not returned from the group buffer pool and had to be read from DASD.

ASYNC READS - DATA RETURND

Total number of asynchronous coupling facility read requests required during this statistics interval. Data was returned from the coupling facility.

ASYNC READS - NO DATA RETURND

Total number of asynchronous coupling facility read requests required during this statistics interval. Data was not returned from the group buffer pool and had to be read from DASD.

PAGES WRITTEN - SYNC + ASYNC - CHANGED

Total number of changed pages written to the group buffer pool during this statistics interval. This value includes both synchronous and asynchronous writes.

PAGES WRITTEN - SYNC + ASYNC - CLEAN

Total number of clean pages written to the group buffer pool during this statistics interval. This value includes both synchronous and asynchronous writes. DB2 writes clean pages for page sets and partitions defined with GBPCACHE=ALL.

OTHER REQUEST

Total number of *other* coupling facility requests not counted in any of the other fields during this statistics interval.

RPL REQUEST

Number of requests to register a page list in the coupling facility (DB2 4.1 and later only).

PG READ AFT RPL

Number of coupling facility reads to retrieve a clean or changed page from the group buffer pool as a result of feedback from the request to register a page list (DB2 4.1 and later only).

DELETE NAME

Number of group buffer pool requests to delete all directory and data entries for a page set or partition (DB2 5.1 and later only).

# STGBp2—DB2 Statistics Global BPool Activity Extended Report

The DB2 Statistics Global BPool Activity Extended Report provides information about the activity of all DB2 global buffer pools, which are grouped by

| | |
|---|---|
| ALL | STGBA2 |
| x | STGBX2 |

By default, STGBX2 reports about global buffer pool GBP0. For a different, single buffer pool, customize a copy of BBPARM member STGBX2 as follows:

- WHERE clause of SQL statement

  Change `BPNAME = 'BP0 '` to `BPNAME = BPx  '`, where `x` is the buffer pool ID of the single buffer pool to be reported (BP0-BP49, BP8K-BP8K9, BP16K-BP16K9, BP32K-BP32K9).

- Title line

  Change (`BP0`) to (`BPx`), where `x` is the buffer pool ID of the single buffer pool to be reported.

```
BMC SOFTWARE INC                   DB2 STATISTICS GLOBAL BPOOL (ALL) ACTIVITY EXTENDED REPORT                         PAGE -     1
REPORT: STGBA2                                                                                       REPORT DATE: 2001-03-25 10.13.42
DATE FROM: 2001-03-01  10.09.44                                                                           LOCATION - DB2J
DATE TO:   2001-03-19  20.52.26                                                                           SUBSYSTEM - DB1J

                                   <-----------SECONDARY------------>
                     CREATE EXPLCT WRITE  WRITE  DELETE DELETE RDCAST DEPEND <-ASYNC REQ-> <-PG PLOCK-->
 DATETIME            THDS   XINVAL  REQS   FAIL  NAMLST  NAME  STATS  GETPGS PRIMRY SECDRY  LOCK  SUSPND
 ------------------- ------ ------ ------ ------ ------ ------ ------ ------ ------ ------ ------ ------
 2001-03-01-10.09.44      0      0      0      0      0      0      0      0      0      0      0      0
 2001-03-01-10.40.30     10      0      0      0      0      0      0      0      0      0      0      0
 2001-03-01-11.10.23      0      0      0      0      0      0      0      0      0      0      0      0
 2001-03-01-11.40.16      0      0      0      0      0      0      0      0      0      0      0      0
 2001-03-01-12.10.10      1      0      0      0      0      0      0      0      0      0      0      0
 2001-03-01-12.40.03      0      0      0      0      0      0      0      0      0      0      0      0
 2001-03-01-13.09.56      1      0      0      0      0      0      0      0      0      0      0      0
 2001-03-01-13.39.49      0      0      0      0      0      0      0      0      0      0      0      0
 2001-03-01-14.09.42      0      0      0      0      0      0      0      0      0      0      0      0
 2001-03-01-14.39.35      0      0      0      0      0      0      0      0      0      0      0      0
 2001-03-01-15.09.28      0      0      0      0      0      0      0      0      0      0      0      0
 2001-03-01-15.39.21      1      0      0      0      0      0      0      0      0      0      0      0
```

Figure 71. DB2 Statistics Global BPool Activity Extended Report

DATETIME
: Date and time that this statistics interval ended.

CREATE THDS
: Number of threads created in this statistics interval.

  **Note:** This is not applicable for the STGBX2 report.

EXPLCT XINVAL
: Number of explicit cross-invalidations (DB2 6.1 and later only).

SECONDARY WRITE REQS
: Number of coupling facility requests to write changed pages to the secondary group buffer pool for duplexing (DB2 6.1 and later only).

SECONDARY WRITE FAIL

Number of coupling facility requests to write changed pages to the secondary group buffer pool for duplexing that failed due to a lack of storage in the coupling facility (DB2 6.1 and later only).

SECONDARY DELETE NAMLST

Number of group buffer pool requests to the secondary group buffer pool to delete a list of pages after they have been cast out from the primary group buffer pool (DB2 6.1 and later only).

SECONDARY DELETE NAME

Number of group buffer pool requests to delete a page from the secondary group buffer pool. Issued by group buffer pool structure owner to delete orphaned data entries in the secondary as part of the garbage collection logic (DB2 6.1 and later only).

SECONDARY RDCAST STATS

Number of coupling facility requests to read the castout statistics for the secondary group buffer pool. Issued by group buffer pool structure owner to check for orphaned data entries in the secondary (DB2 6.1 and later only).

DEPEND GETPGS

Number of getpages for global buffer pool dependent pages (DB2 7.1 and later only).

ASYNC REQ-PRIMRY

Number of asynchronous IXLCACHE invocations for the primary group buffer pool (DB2 6.1 and later only).

ASYNC REQ-SECDRY

Number of asynchronous IXLCACHE invocations for the secondary group buffer pool (DB2 6.1 and later only).

PG PLOCK-LOCK

Number of page P-lock lock requests for space map, data, and index leaf pages (DB2 7.1 and later only).

PG PLOCK-SUSPND

Number of page P-lock lock suspensions for space map, data, and index leaf pages (DB2 7.1 and later only).

## STGBpEX—DB2 Statistics Global BPool Exceptions Report

The DB2 Statistics Global BPool Exceptions Report provides information about the exceptions activity of all DB2 global buffer pools, which are grouped by

| | |
|---|---|
| 4K | STGB4EX |
| 32K | STGB3EX |
| ALL | STGBAEX |
| x | STGBXEX |

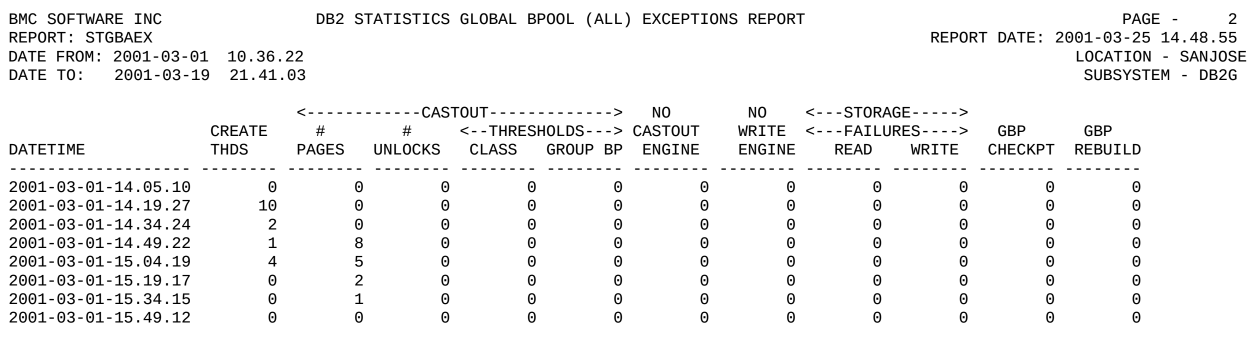
By default, STGBXEX reports about global buffer pool GBP0. For a different, single buffer pool, customize a copy of BBPARM member STGBXEX as follows:

- WHERE clause of SQL statement

  Change `BPNAME = 'BP0 '` to `BPNAME = BPx  '`, where `x` is the buffer pool ID of the single buffer pool to be reported (BP0-BP49, BP8K-BP8K9, BP16K-BP16K9, BP32K-BP32K9).

- Title line

  Change `(BP0)` to `(BPx)`, where `x` is the buffer pool ID of the single buffer pool to be reported.



```
BMC SOFTWARE INC                DB2 STATISTICS GLOBAL BPOOL (ALL) EXCEPTIONS REPORT                                PAGE -     2
REPORT: STGBAEX                                                                                REPORT DATE: 2001-03-25 14.48.55
DATE FROM: 2001-03-01  10.36.22                                                                              LOCATION - SANJOSE
DATE TO:   2001-03-19  21.41.03                                                                               SUBSYSTEM - DB2G

                              <------------CASTOUT------------->   NO        NO    <---STORAGE----->
                     CREATE     #        #     <--THRESHOLDS---> CASTOUT    WRITE  <---FAILURES---->   GBP      GBP
DATETIME             THDS     PAGES   UNLOCKS   CLASS   GROUP BP  ENGINE    ENGINE    READ    WRITE   CHECKPT  REBUILD
------------------- -------- -------- -------- -------- -------- -------- -------- -------- -------- -------- --------
2001-03-01-14.05.10        0        0        0        0        0        0        0        0        0        0        0
2001-03-01-14.19.27       10        0        0        0        0        0        0        0        0        0        0
2001-03-01-14.34.24        2        0        0        0        0        0        0        0        0        0        0
2001-03-01-14.49.22        1        8        0        0        0        0        0        0        0        0        0
2001-03-01-15.04.19        4        5        0        0        0        0        0        0        0        0        0
2001-03-01-15.19.17        0        2        0        0        0        0        0        0        0        0        0
2001-03-01-15.34.15        0        1        0        0        0        0        0        0        0        0        0
2001-03-01-15.49.12        0        0        0        0        0        0        0        0        0        0        0
```

Figure 72. DB2 Statistics Global BPool Exceptions Report

DATETIME
Date and time that this statistics interval ended.

CREATE THDS
Number of threads created in this statistics interval.

**Note:** This is not applicable for the STGBXEX report.

CASTOUT - # PAGES
Number of pages cast out from the group buffer pool to DASD during this statistics interval.

CASTOUT - # UNLOCKS
Number of coupling facility requests to unlock the castout lock on the pages (DB2 5.1 and later only).

CASTOUT - THRESHOLDS - CLASS
: Number of times group buffer pool castout was initiated because the class castout threshold was detected during this statistics interval.

CASTOUT - THRESHOLDS - GROUP BP
: Number of times group buffer pool castout was initiated because the group buffer pool castout threshold was detected during this statistics interval.

NO CASTOUT ENGINE
: Number of times a castout engine was not available during this statistics interval.

NO WRITE ENGINE
: Number of times a coupling facility write engine was not available for coupling facility writes during this statistics interval.

STORAGE FAILURES - READ
: Number of coupling facility read requests not completed because of a lack of coupling facility storage resources during this statistics interval.

STORAGE FAILURES - WRITE
: Number of coupling facility write requests not completed because of a lack of coupling facility storage resources during this statistics interval.

GBP CHECKPT
: Number of group buffer pool checkpoints triggered by this member (DB2 5.1 and later only).

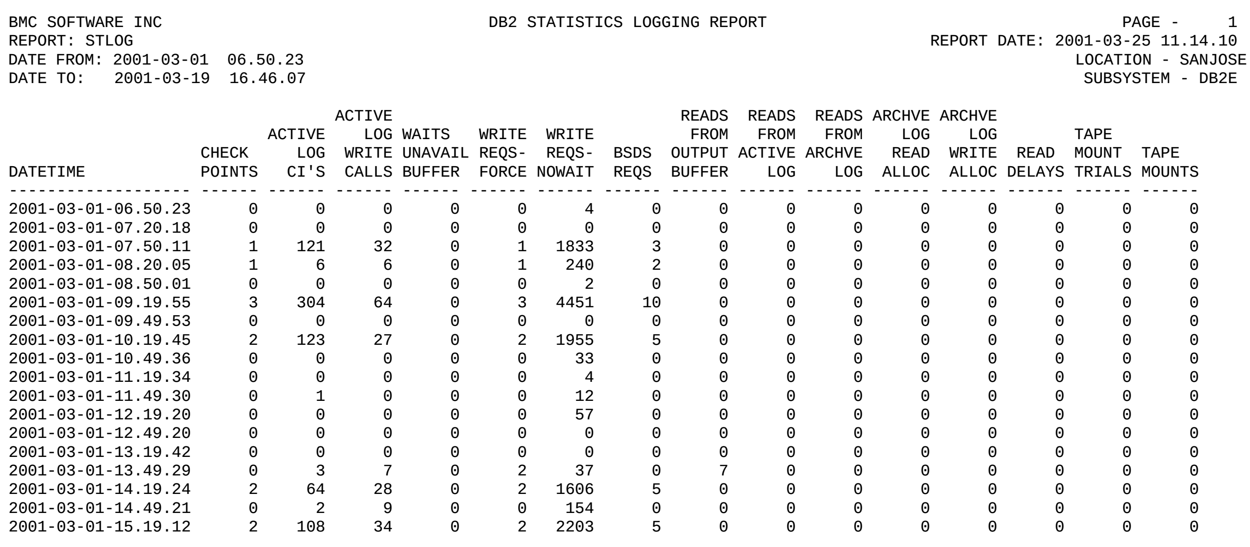
GBP REBUILD
: Number of group buffer pool rebuilds in which this member participated (DB2 5.1 and later only).

# Log Reports

For a statistics interval, the log reports provide totals about the amount of active log activity, log request waits, log write requests, bootstrap data set requests, log reads, log writes, read delays, and tape mounts.

# STLOG—DB2 Statistics Logging Report

The DB2 Statistics Logging Report displays information on log status and activity.



```
BMC SOFTWARE INC                                  DB2 STATISTICS LOGGING REPORT                                    PAGE -     1
REPORT: STLOG                                                                                   REPORT DATE: 2001-03-25 11.14.10
DATE FROM: 2001-03-01  06.50.23                                                                                LOCATION - SANJOSE
DATE TO:   2001-03-19  16.46.07                                                                                SUBSYSTEM - DB2E

                                 ACTIVE                              READS  READS  READS ARCHVE ARCHVE
                          ACTIVE    LOG WAITS   WRITE  WRITE          FROM   FROM   FROM    LOG    LOG        TAPE
                    CHECK    LOG  WRITE UNAVAIL REQS-  REQS-   BSDS OUTPUT ACTIVE ARCHVE   READ  WRITE  READ  MOUNT  TAPE
DATETIME           POINTS   CI'S  CALLS BUFFER  FORCE NOWAIT   REQS BUFFER    LOG    LOG  ALLOC  ALLOC DELAYS TRIALS MOUNTS
------------------- ------ ------ ------ ------ ------ ------ ------ ------ ------ ------ ------ ------ ------ ------ ------
2001-03-01-06.50.23      0      0      0      0      0      4      0      0      0      0      0      0      0      0      0
2001-03-01-07.20.18      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0
2001-03-01-07.50.11      1    121     32      0      1   1833      3      0      0      0      0      0      0      0      0
2001-03-01-08.20.05      1      6      6      0      1    240      2      0      0      0      0      0      0      0      0
2001-03-01-08.50.01      0      0      0      0      0      2      0      0      0      0      0      0      0      0      0
2001-03-01-09.19.55      3    304     64      0      3   4451     10      0      0      0      0      0      0      0      0
2001-03-01-09.49.53      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0
2001-03-01-10.19.45      2    123     27      0      2   1955      5      0      0      0      0      0      0      0      0
2001-03-01-10.49.36      0      0      0      0      0     33      0      0      0      0      0      0      0      0      0
2001-03-01-11.19.34      0      0      0      0      0      4      0      0      0      0      0      0      0      0      0
2001-03-01-11.49.30      0      1      0      0      0     12      0      0      0      0      0      0      0      0      0
2001-03-01-12.19.20      0      0      0      0      0     57      0      0      0      0      0      0      0      0      0
2001-03-01-12.49.20      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0
2001-03-01-13.19.42      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0
2001-03-01-13.49.29      0      3      7      0      2     37      0      7      0      0      0      0      0      0      0
2001-03-01-14.19.24      2     64     28      0      2   1606      5      0      0      0      0      0      0      0      0
2001-03-01-14.49.21      0      2      9      0      0    154      0      0      0      0      0      0      0      0      0
2001-03-01-15.19.12      2    108     34      0      2   2203      5      0      0      0      0      0      0      0      0
```

Figure 73. DB2 Statistics Logging Report

DATETIME
: Date and time that this statistics interval ended.

CHECK POINTS
: Number of DB2 checkpoints taken during this statistics interval.

ACTIVE LOG CI’S
: Total number of active log output control intervals (CIs) created during this statistics interval.

ACTIVE LOG WRITE CALLS
: Total number of calls to write active log buffers during this statistics interval.

WAITS UNAVAIL BUFFER
: Total number of times log requests had to wait because of unavailable output buffers during this statistics interval.

WRITE REQS-FORCE
: Total number of FORCE log-write requests during this statistics interval.

WRITE REQS-NOWAIT
: Total number of NOWAIT log-write requests during this statistics interval.

BSDS REQS.
Total number of bootstrap data set requests during this statistics interval.

READS FROM OUTPUT BUFFER
Total number of reads from the log output buffer for backout during this statistics interval.

READS FROM ACTIVE LOG
Total number of reads from the active log for backout during this statistics interval.

READS FROM ARCHIVE LOG
Total number of reads from the archive log for backout during this statistics interval.

ARCHIVE LOG READ ALLOCS
Total number of archive log read allocations during this statistics interval.

ARCHIVE LOG WRITE ALLOCS
Total number of archive log write allocations during this statistics interval.

READ DELAYS
Total number of reads delayed during this statistics interval. A read can be delayed because of tape volume contention or unavailable resources (usually, allocated tape units) during this statistics interval.

TAPE MOUNT TRIALS
Total number of ‘look-ahead’ tape mounts attempted during this statistics interval.

TAPE MOUNTS
Total number of successful ‘look-ahead’ tape mounts during this statistics interval. The difference between TAPE MOUNT TRIALS and this number is the number of failed attempts.

# Chapter 15. Audit Reports

This chapter describes the audit reports produced by Performance Reporter. Each of the following reports is defined in a separate member in BBPARM:

- The DB2 Audit Summary Report (classes 1 - 8)
  - AUSUM
- The DB2 Authorization Failures Report (class 1)
  - AUFAIL
- The DB2 Authorization Control—GRANTs / REVOKEs Report (class 2)
  - AUDGRV
- The DB2 Audited DDL Access Report (class 3)
  - AUDDL
- The DB2 Audited DML Access Report (classes 4 and 5)
  - AUDML
- The DB2 DML at BIND Report (class 6)
  - AUDMLB
- The DB2 AUTHID Change Report (class 7)
  - AUCHNG
- The DB2 Audit Utility Access Report (class 8)
  - AUUTIL
- The DB2 Audit Detail Report (classes 1 - 8)
  - AUDTL

Rename or comment out the corresponding report requests from job DPRREPT for those audit classes not processed.

# AUSUM—DB2 Audit Summary Report

The DB2 Audit Summary Report summarizes the audit records for each authorization ID and plan. It requires one or more of the audit classes 1 through 8 (TYPE2=AUDSUM).

```
BMC SOFTWARE INC                                    DB2 AUDIT SUMMARY REPORT                               PAGE -     1
REPORT: AUSUM                                                                          REPORT DATE: 2001-03-25 19.25.28
DATE FROM: 2001-03-01 11.06.09                                                                           SYSTEM - SYSB
DATE TO:   2001-03-19 11.51.34                                                                        SUBSYSTEM - DB2E

                                                          DML      DML
                    TOTAL      AUTH   GRANT/    DDL      READ     WRITE   DML AT   AUTHID   UTILITY
 AUTHID    PLAN    AUDITS   FAILURE REVOKE   ACCESS   ACCESS   ACCESS    BIND    CHANGE   ACCESS
 -------- -------- -------- -------- -------- -------- -------- -------- -------- -------- --------
 ACCTDEP                  18        0        0        0        0        0        0       18        0
 ACCTDEP  TSMPL03       2748        0        0        0        0        0        0     2748        0
 CJN1     DSNESPRR        17        0       17        0        0        0        0        0        0
 CJN2     QMF311           2        2        0        0        0        0        0        0        0
 CMR1                      4        0        0        0        0        0        0        4        0
 CMR1     TSMPL04          5        0        0        0        0        0        0        5        0
 ELC2     MXZAPPL          8        0        8        0        0        0        0        0        0
 LAA1     RXSEL1M      15877    15877        0        0        0        0        0        0        0
 MXW2     DSNUTIL         28        0        0        0        4        2        0        0       22
 MXW2     MWSUMLD         33        0        0        0        0        8       25        0        0
 MXW2     RXDB2            5        0        0        0        0        0        5        0        0
 MXW3     DSNUTIL          2        0        0        0        0        0        0        0        2
 MXW3     MWSUMLD         10        0        0        0        0        2        8        0        0
 PAYROLDP                  1        0        0        0        0        0        0        1        0
 PAYROLDP TSMPL02          1        0        0        0        0        0        0        1        0
 SYSOPR                  459        3        0        0        0        0        0      456        0
 TSM5                      3        0        0        0        0        0        0        3        0
 TSM5     TSMPL05          4        0        0        0        0        0        0        4        0
 TSM6                      1        0        0        0        0        0        0        1        0
 TSM6     TSMPL06          1        0        0        0        0        0        0        1        0
 TSM7                      2        0        0        0        0        0        0        2        0
 TSM7     TSMPL07          3        0        0        0        0        0        0        3        0
 TSM8                      3        0        0        0        0        0        0        3        0
 TSM8     TSMPL08          6        0        0        0        0        0        0        6        0
```

Figure 74. DB2 Audit Summary Report

The report data is sorted by both AUTHID and PLAN (default).

AUTHID
: Authorization ID of the user accessing this DB2 system.

PLAN
: Plan name of the application program, transaction, or utility executed.

TOTAL AUDITS
: Number of events from all selected audit types.

AUTH FAILURE
: Number of authorization failures.

GRANT / REVOKE
: Number of authorization GRANTs or REVOKEs.

DDL ACCESS
: Number of DDL operations against audited DB2 tables.

DML READ ACCESS
: Number of first READ attempts within a logical unit of work against audited DB2 tables.

DML WRITE ACCESS
: Number of first WRITE attempts within a logical unit of work against audited DB2 tables.

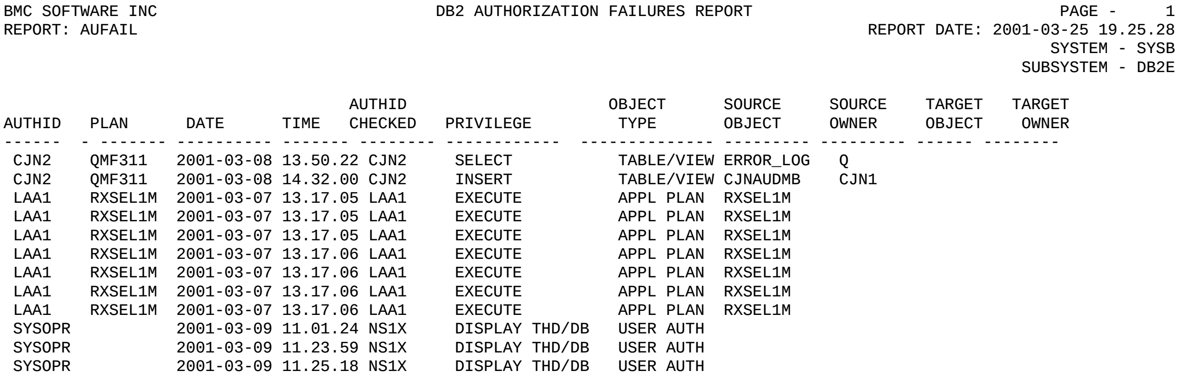
DML AT BIND
Number of DML statements referenced during a static or dynamic BIND against audited DB2 tables.

AUTHID CHANGE
Number of initial authorization ID establishments, changes, or attempted changes.

UTILITY ACCESS
Number of utility trace records written against DB2 tables.

## AUFAIL—DB2 Authorization Failures Report

The DB2 Authorization Failures Report provides data on authorization failures for each authorization ID and plan. It identifies each user who attempted to access a DB2 object and failed due to lack of authorization. It requires audit class 1 (TYPE2=AUDFAIL).



```
BMC SOFTWARE INC                             DB2 AUTHORIZATION FAILURES REPORT                                PAGE -     1
REPORT: AUFAIL                                                                            REPORT DATE: 2001-03-25 19.25.28
                                                                                                             SYSTEM - SYSB
                                                                                                          SUBSYSTEM - DB2E

                                    AUTHID                     OBJECT      SOURCE     SOURCE    TARGET   TARGET
AUTHID   PLAN      DATE      TIME   CHECKED   PRIVILEGE         TYPE       OBJECT     OWNER     OBJECT    OWNER
------  - ------- ---------- ------- -------- ------------  -------------- --------- --------- ------ --------
 CJN2    QMF311   2001-03-08 13.50.22 CJN2     SELECT           TABLE/VIEW ERROR_LOG   Q
 CJN2    QMF311   2001-03-08 14.32.00 CJN2     INSERT           TABLE/VIEW CJNAUDMB    CJN1
 LAA1    RXSEL1M  2001-03-07 13.17.05 LAA1     EXECUTE          APPL PLAN  RXSEL1M
 LAA1    RXSEL1M  2001-03-07 13.17.05 LAA1     EXECUTE          APPL PLAN  RXSEL1M
 LAA1    RXSEL1M  2001-03-07 13.17.05 LAA1     EXECUTE          APPL PLAN  RXSEL1M
 LAA1    RXSEL1M  2001-03-07 13.17.06 LAA1     EXECUTE          APPL PLAN  RXSEL1M
 LAA1    RXSEL1M  2001-03-07 13.17.06 LAA1     EXECUTE          APPL PLAN  RXSEL1M
 LAA1    RXSEL1M  2001-03-07 13.17.06 LAA1     EXECUTE          APPL PLAN  RXSEL1M
 LAA1    RXSEL1M  2001-03-07 13.17.06 LAA1     EXECUTE          APPL PLAN  RXSEL1M
 SYSOPR           2001-03-09 11.01.24 NS1X     DISPLAY THD/DB   USER AUTH
 SYSOPR           2001-03-09 11.23.59 NS1X     DISPLAY THD/DB   USER AUTH
 SYSOPR           2001-03-09 11.25.18 NS1X     DISPLAY THD/DB   USER AUTH
```

Figure 75. DB2 Authorization Failures Report

The report data is sorted by both AUTHID and PLAN (default).

AUTHID
: Authorization ID of the user accessing this DB2 system.

PLAN
: Plan name of the application program, transaction, or utility executed.

DATE
: Date of the authorization failure.

TIME
: Time of the authorization failure.

AUTHID CHECKED
: Authorization ID that caused the failure.

PRIVILEGE
: Operation (privilege) on which the failure occurred, such as SELECT, INSERT, BIND, ADD, or LOCK TABLE.

OBJECT TYPE
: DB2 object type, such as BUFFERPOOL, PLAN, or TABLESPACE.

SOURCE OBJECT
: Source DB2 table name or plan name involved in the authorization failure.

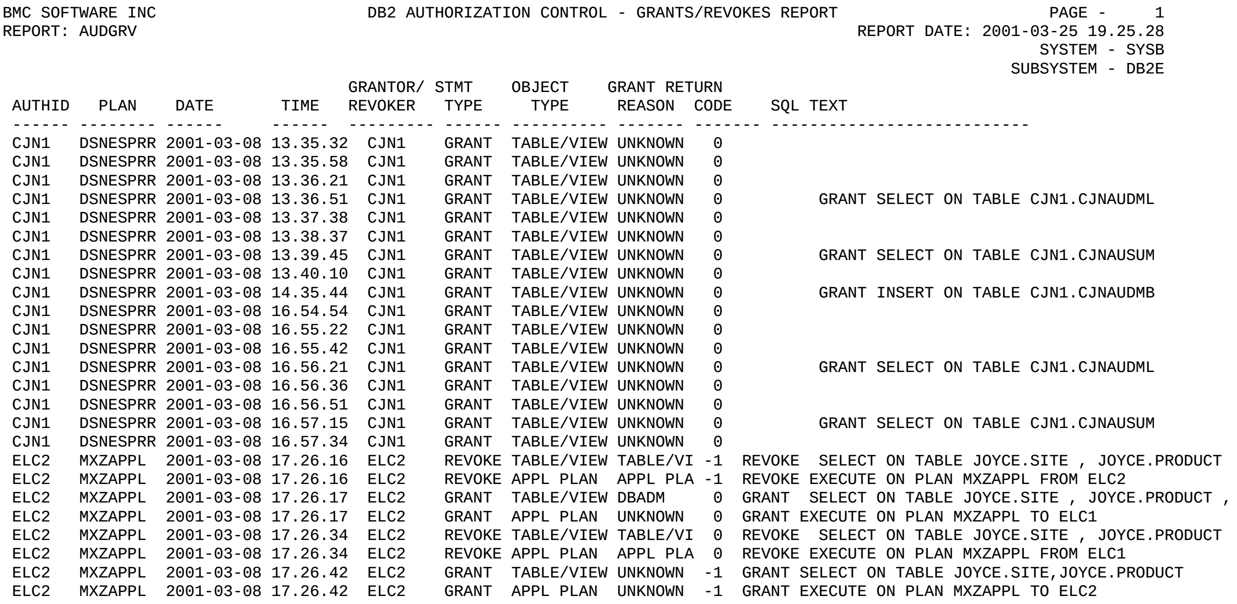
SOURCE OWNER
: Authorization ID of the owner of the source DB2 table.

TARGET OBJECT
: Target DB2 table name or plan name involved in the authorization failure.

TARGET OWNER
: Authorization ID of the owner of the target DB2 table.

# AUDGRV—DB2 Authorization Control—GRANTs / REVOKEs Report

The DB2 Authorization Control Report provides data on all GRANT and REVOKE statements for each authorization ID and plan. It requires audit class 2 (TYPE2=AUDGRV).



```
BMC SOFTWARE INC                      DB2 AUTHORIZATION CONTROL - GRANTS/REVOKES REPORT                      PAGE -     1
REPORT: AUDGRV                                                                           REPORT DATE: 2001-03-25 19.25.28
                                                                                                            SYSTEM - SYSB
                                                                                                         SUBSYSTEM - DB2E
                                    GRANTOR/ STMT    OBJECT    GRANT RETURN
 AUTHID   PLAN    DATE       TIME   REVOKER   TYPE     TYPE     REASON  CODE    SQL TEXT
 ------ -------- ------     ------  --------- ------ ---------- ------- ------- --------------------------
 CJN1   DSNESPRR 2001-03-08 13.35.32  CJN1    GRANT  TABLE/VIEW UNKNOWN   0
 CJN1   DSNESPRR 2001-03-08 13.35.58  CJN1    GRANT  TABLE/VIEW UNKNOWN   0
 CJN1   DSNESPRR 2001-03-08 13.36.21  CJN1    GRANT  TABLE/VIEW UNKNOWN   0
 CJN1   DSNESPRR 2001-03-08 13.36.51  CJN1    GRANT  TABLE/VIEW UNKNOWN   0          GRANT SELECT ON TABLE CJN1.CJNAUDML
 CJN1   DSNESPRR 2001-03-08 13.37.38  CJN1    GRANT  TABLE/VIEW UNKNOWN   0
 CJN1   DSNESPRR 2001-03-08 13.38.37  CJN1    GRANT  TABLE/VIEW UNKNOWN   0
 CJN1   DSNESPRR 2001-03-08 13.39.45  CJN1    GRANT  TABLE/VIEW UNKNOWN   0          GRANT SELECT ON TABLE CJN1.CJNAUSUM
 CJN1   DSNESPRR 2001-03-08 13.40.10  CJN1    GRANT  TABLE/VIEW UNKNOWN   0
 CJN1   DSNESPRR 2001-03-08 14.35.44  CJN1    GRANT  TABLE/VIEW UNKNOWN   0          GRANT INSERT ON TABLE CJN1.CJNAUDMB
 CJN1   DSNESPRR 2001-03-08 16.54.54  CJN1    GRANT  TABLE/VIEW UNKNOWN   0
 CJN1   DSNESPRR 2001-03-08 16.55.22  CJN1    GRANT  TABLE/VIEW UNKNOWN   0
 CJN1   DSNESPRR 2001-03-08 16.55.42  CJN1    GRANT  TABLE/VIEW UNKNOWN   0
 CJN1   DSNESPRR 2001-03-08 16.56.21  CJN1    GRANT  TABLE/VIEW UNKNOWN   0          GRANT SELECT ON TABLE CJN1.CJNAUDML
 CJN1   DSNESPRR 2001-03-08 16.56.36  CJN1    GRANT  TABLE/VIEW UNKNOWN   0
 CJN1   DSNESPRR 2001-03-08 16.56.51  CJN1    GRANT  TABLE/VIEW UNKNOWN   0
 CJN1   DSNESPRR 2001-03-08 16.57.15  CJN1    GRANT  TABLE/VIEW UNKNOWN   0          GRANT SELECT ON TABLE CJN1.CJNAUSUM
 CJN1   DSNESPRR 2001-03-08 16.57.34  CJN1    GRANT  TABLE/VIEW UNKNOWN   0
 ELC2   MXZAPPL  2001-03-08 17.26.16  ELC2    REVOKE TABLE/VIEW TABLE/VI -1  REVOKE  SELECT ON TABLE JOYCE.SITE , JOYCE.PRODUCT
 ELC2   MXZAPPL  2001-03-08 17.26.16  ELC2    REVOKE APPL PLAN  APPL PLA -1  REVOKE EXECUTE ON PLAN MXZAPPL FROM ELC2
 ELC2   MXZAPPL  2001-03-08 17.26.17  ELC2    GRANT  TABLE/VIEW DBADM     0  GRANT  SELECT ON TABLE JOYCE.SITE , JOYCE.PRODUCT ,
 ELC2   MXZAPPL  2001-03-08 17.26.17  ELC2    GRANT  APPL PLAN  UNKNOWN   0  GRANT EXECUTE ON PLAN MXZAPPL TO ELC1
 ELC2   MXZAPPL  2001-03-08 17.26.34  ELC2    REVOKE TABLE/VIEW TABLE/VI  0  REVOKE  SELECT ON TABLE JOYCE.SITE , JOYCE.PRODUCT
 ELC2   MXZAPPL  2001-03-08 17.26.34  ELC2    REVOKE APPL PLAN  APPL PLA  0  REVOKE EXECUTE ON PLAN MXZAPPL FROM ELC1
 ELC2   MXZAPPL  2001-03-08 17.26.42  ELC2    GRANT  TABLE/VIEW UNKNOWN  -1  GRANT SELECT ON TABLE JOYCE.SITE,JOYCE.PRODUCT
 ELC2   MXZAPPL  2001-03-08 17.26.42  ELC2    GRANT  APPL PLAN  UNKNOWN  -1  GRANT EXECUTE ON PLAN MXZAPPL TO ELC2
```

Figure 76. DB2 Authorization Control—GRANTs / REVOKEs Report

The report data is sorted by both AUTHID and PLAN (default).

AUTHID
: Authorization ID of the user accessing this DB2 system.

PLAN
: Plan name of the application program, transaction, or utility executed.

DATE
: Date of the GRANT or REVOKE operation.

TIME
: Time of the GRANT or REVOKE operation.

GRANTOR / REVOKER
: Authorization ID of the user issuing the GRANT or REVOKE.

STMT TYPE
: Statement type—either GRANT or REVOKE.

OBJECT TYPE
: DB2 object type of the GRANT or REVOKE, such as BUFFERPOOL, PLAN, or TABLESPACE.

GRANT REASON
Authorization level of the grantor: SYSADM, SYSCTRL, SYSOPR, DBADM, DBCTRL, or DBMAINT.

RETURN CODE
SQL return code from the GRANT / REVOKE operation.

SQL TEXT
Beginning of the text of the GRANT or REVOKE statement.

## AUDDL—DB2 Audited DDL Access Report

The DB2 Audited DDL Access Report provides DDL data for each authorization ID and plan. It identifies DDL status changes (CREATE, DROP, ALTER) on audited tables. It requires audit class 3 (TYPE2=AUDDDL).

```
BMC SOFTWARE INC                                   DB2 AUDITED DDL ACCESS REPORT                                      PAGE -     1
REPORT: AUDDL                                                                                    REPORT DATE: 2001-03-25 16.53.08
                                                                                                                    SYSTEM - SYSB
                                                                                                                 SUBSYSTEM - DB2E
                                                                   TABLE     TABLE  TABLE
 AUTHID    PLAN     DATE       TIME     TYPE      TABLE NAME       OWNER    CREATOR OBID  DATABASE SQL TEXT
-------- -------- ---------- -------- -------- ----------------- -------- -------- ----- -------- -------------------------------
MXW3    DSNESPCS 2001-03-07 22.08.32 DROP     DMRAUSUM          MXW2     MXW2         96              DROP TABLESPACE MWDMRDB1
MXW3    DSNESPCS 2001-03-07 22.09.06 DROP     DMRAUSUM          MXW2     MXW2         96              DROP TABLESPACE MWDMRDB1
MXW3    DSNESPCS 2001-03-07 22.09.22 CREATE   DMRAUSUM          MXW2     MXW3         96
MXW3    DSNESPCS 2001-03-07 22.30.48 DROP     DMRAUSUM          MXW2     MXW3         96              DROP TABLESPACE MWDMRDB1
MXW3    DSNESPCS 2001-03-07 22.31.02 CREATE   DMRAUSUM          MXW2     MXW3         96
MXW3    DSNESPCS 2001-03-07 22.48.41 DROP     DMRAUDMB          MXW2     MXW2         71              DROP TABLESPACE MWDMRDB1
MXW3    DSNESPCS 2001-03-07 22.48.41 DROP     DMRAUCHG          MXW2     MXW2          3              DROP TABLESPACE MWDMRDB1
MXW3    DSNESPCS 2001-03-07 22.48.41 DROP     DMRAUDDL          MXW2     MXW2         76              DROP TABLESPACE MWDMRDB1
MXW3    DSNESPCS 2001-03-07 22.48.42 DROP     DMRAUDML          MXW2     MXW2         81              DROP TABLESPACE MWDMRDB1
MXW3    DSNESPCS 2001-03-07 22.48.42 DROP     DMRAUGRV          MXW2     MXW2         91              DROP TABLESPACE MWDMRDB1
MXW3    DSNESPCS 2001-03-07 22.48.42 DROP     DMRAUFAL          MXW2     MXW2         86              DROP TABLESPACE MWDMRDB1
MXW3    DSNESPCS 2001-03-07 22.48.42 DROP     DMRAUUTL          MXW2     MXW2        101              DROP TABLESPACE MWDMRDB1
MXW3    DSNESPCS 2001-03-07 22.48.42 DROP     DMRAUSUM          MXW2     MXW3         96              DROP TABLESPACE MWDMRDB1
MXW3    DSNESPCS 2001-03-07 22.49.30 CREATE   DMRAUCHG          MXW2     MXW3          3
MXW3    DSNESPCS 2001-03-07 22.49.34 CREATE   DMRAUDMB          MXW2     MXW3         71
MXW3    DSNESPCS 2001-03-07 22.49.41 CREATE   DMRAUDDL          MXW2     MXW3         76
MXW3    DSNESPCS 2001-03-07 22.50.11 CREATE   DMRAUCHG          MXW2     MXW3          3
MXW3    DSNESPCS 2001-03-07 22.50.15 CREATE   DMRAUDMB          MXW2     MXW3         71
MXW3    DSNESPCS 2001-03-07 22.50.21 CREATE   DMRAUDDL          MXW2     MXW3         76
MXW3    DSNESPCS 2001-03-07 22.51.33 CREATE   DMRAUCHG          MXW2     MXW3          3
MXW3    DSNESPCS 2001-03-07 22.51.37 CREATE   DMRAUDMB          MXW2     MXW3         71
MXW3    DSNESPCS 2001-03-07 22.51.41 CREATE   DMRAUDDL          MXW2     MXW3         76
MXW3    DSNESPCS 2001-03-07 22.51.46 CREATE   DMRAUDML          MXW2     MXW3         81
MXW3    DSNESPCS 2001-03-07 22.51.51 CREATE   DMRAUFAL          MXW2     MXW3         86              CREATE TABLE MXW2.DMRAUFAL
MXW3    DSNESPCS 2001-03-07 22.51.56 CREATE   DMRAUGRV          MXW2     MXW3         91
MXW3    DSNESPCS 2001-03-07 22.52.01 CREATE   DMRAUSUM          MXW2     MXW3         96
MXW3    DSNESPCS 2001-03-07 22.52.07 CREATE   DMRAUUTL          MXW2     MXW3        101
MXW3    DSNESPCS 2001-03-08 14.12.28 DROP     DMRAUDDL          MXW2     MXW3         76              DROP TABLESPACE MWDMRDB1
MXW3    DSNESPCS 2001-03-08 14.13.12 CREATE   DMRAUDDL          MXW2     MXW3         76
```

Figure 77. DB2 Audited DDL Access Report

The report data is sorted by both AUTHID and PLAN (default).

AUTHID
: Authorization ID of the user accessing this DB2 system.

PLAN
: Plan name of the application program, transaction, or utility executed.

DATE
: Date of the audited DDL access.

TIME
: Time of the audited DDL access.

TYPE
: Type of access performed on the audited DB2 table, such as CREATE, ALTER, or DROP.

TABLE NAME
: Name of the accessed audited DB2 table.

TABLE OWNER
: User ID of the owner of the audited DB2 table accessed.

TABLE CREATOR
Authorization ID of the creator of the DB2 table associated with the DDL access.

TABLE OBID
Internal DB2 identification (OBID) of the table associated with the access.

DATABASE
Name of the database (or DBID) containing the audited DB2 table.

SQL TEXT
Beginning of the text of the SQL statement accessing the audited DB2 table.

## AUDML—DB2 Audited DML Access Report

The DB2 Audited DML Access Report contains DML data for each authorization ID and plan. It identifies DML accesses to audited tables. It requires audit classes 4 and/or 5 (TYPE2= AUDDML).

```
BMC SOFTWARE INC                               DB2 AUDITED DML ACCESS REPORT                                PAGE -     1
REPORT: AUDML                                                                                   REPORT DATE: 2001-03-25 19.25.28
                                                                                                               SYSTEM - SYSB
                                                                                                            SUBSYSTEM - DB2E
AUTHID    PLAN      DATE      TIME    STMT TYPE TABLE DATABASE PAGESET LOG RBA(HEX)
-------- -------- ---------- -------- ---------- ----- -------- -------- ------------
MXW2     DSNUTIL 2001-03-08 07.09.20 1ST READ       8 MWDMRDB1 DMRPRTAS 000800000D43
MXW2     DSNUTIL 2001-03-08 07.17.16 1ST READ       8 MWDMRDB1 DMRPRTAS 000800000D44
MXW2     DSNUTIL 2001-03-08 08.36.12 1ST READ       8 MWDMRDB1 DMRPRTAS 000800000D45
MXW2     DSNUTIL 2001-03-08 08.36.57 1ST WRITE      8 MWDMRDB1 DMRPRTAS 000800000D45
MXW2     DSNUTIL 2001-03-08 08.37.04 1ST READ       8 MWDMRDB1 DMRPRTAS 000800000D45
MXW2     DSNUTIL 2001-03-08 08.37.04 1ST WRITE      8 MWDMRDB1 DMRPRTAS 000800000D45
MXW2     MWSUMLD 2001-03-07 14.41.07 1ST WRITE     23 MWDMRDB1 DMRPRTDS 001700000000
MXW2     MWSUMLD 2001-03-08 07.08.51 1ST WRITE      8 MWDMRDB1 DMRPRTAS 000800000D18
MXW2     MWSUMLD 2001-03-08 07.09.14 1ST WRITE     23 MWDMRDB1 DMRPRTDS 001700000000
MXW2     MWSUMLD 2001-03-08 07.17.09 1ST WRITE     23 MWDMRDB1 DMRPRTDS 001700000000
MXW2     MWSUMLD 2001-03-08 07.17.09 1ST WRITE      8 MWDMRDB1 DMRPRTAS 000800000000
MXW2     MWSUMLD 2001-03-08 08.35.52 1ST WRITE      8 MWDMRDB1 DMRPRTAS 000800000D44
MXW2     MWSUMLD 2001-03-08 08.36.03 1ST WRITE     23 MWDMRDB1 DMRPRTDS 001700000000
MXW2     MWSUMLD 2001-03-08 10.50.37 1ST WRITE     23 MWDMRDB1 DMRPRTDS 001700000000
MXW3     MWSUMLD 2001-03-07 17.38.37 1ST WRITE     23 MWDMRDB1 DMRPRTDS 001700000000
MXW3     MWSUMLD 2001-03-08 11.30.01 1ST WRITE     23 MWDMRDB1 DMRPRTDS 001700000000
```

Figure 78. DB2 Audited DML Access Report

The report data is sorted by both AUTHID and PLAN (default).

AUTHID
: Authorization ID of the user accessing this DB2 system.

PLAN
: Plan name of the application program, transaction, or utility executed.

DATE
: Date of the audited DML access.

TIME
: Time of the audited DML access.

STMT TYPE
: Type of access performed on the audited DB2 table.

TABLE
: Internal DB2 identification (OBID) of the table associated with the access.

DATABASE
: Name of the database (or DBID) containing the audited DB2 table.

PAGESET
: Name of the page set (or OBID) containing the audited DB2 table.

LOG RBA (HEX)
: Log relative byte address of the unit of recovery in which the audited DB2 table was accessed.

# AUDMLB—DB2 DML at BIND Report

The DB2 DML at BIND Report provides data on DML statements at BIND time for each authorization ID and plan. It identifies DML statements referenced during a static or dynamic BIND against audited DB2 tables. It requires audit class 6 (TYPE2=AUDDMB).

```
BMC SOFTWARE INC                                   DB2 DML AT BIND REPORT                                             PAGE -     1
REPORT: AUDMLB                                                                                    REPORT DATE: 2001-03-25 19.25.28
                                                                                                                     SYSTEM - SYSB
                                                                                                                  SUBSYSTEM - DB2E

                                                                              SQL
                                      PACKAGE            PROGRAM/ STATEMENT   STMT TABLE DATA PRECOMPILE       RETURN
AUTHID   PLAN      DATE       TIME   COLL ID              DBRM     TYPE       #    OBID BASE TIMESTAMP(HEX)   CODE SQL TEXT
-------- -------- ---------- -------- -------------- ---------- ----- ----- ----- ---------------- ----- -------------
MXW2     MWSUMLD 2001-03-07 12.33.31                 DPSPURGD DELETE        58     8   273 0C2D7C4F00F4F4F2     0 DELETE FROM MX
MXW2     MWSUMLD 2001-03-07 12.33.32                 DPSPURGD DELETE        58    23   273 0C2D7C4F00F4F4F2     0 DELETE FROM MX
MXW2     MWSUMLD 2001-03-07 12.33.35                 DPSQLDAD SEL QUERY    105     8   273 0C2D7C4F00F4F4F2     0 SELECT * FROM
MXW2     MWSUMLD 2001-03-07 12.33.37                 DPSQLDAD SEL QUERY    105     8   273 0C2D7C4F00F4F4F2     0 SELECT * FROM
MXW2     MWSUMLD 2001-03-07 14.41.02                 DPSPURGD DELETE        58     8   273 0C2D7C4F00F4F4F2     0 DELETE FROM MX
MXW2     MWSUMLD 2001-03-07 14.41.03                 DPSPURGD DELETE        58    23   273 0C2D7C4F00F4F4F2     0 DELETE FROM MX
MXW2     MWSUMLD 2001-03-07 14.41.04                 DPSQLDAD SEL QUERY    105     8   273 0C2D7C4F00F4F4F2     0 SELECT * FROM
MXW2     MWSUMLD 2001-03-07 14.41.06                 DPSQLDAD SEL QUERY    105     8   273 0C2D7C4F00F4F4F2     0 SELECT * FROM
MXW2     MWSUMLD 2001-03-08 07.08.19                 DPSPURGD DELETE        58     8   273 0C2D7C4F00F4F4F2     0 DELETE FROM MX
MXW2     MWSUMLD 2001-03-08 07.09.08                 DPSPURGD DELETE        58    23   273 0C2D7C4F00F4F4F2     0 DELETE FROM MX
MXW2     MWSUMLD 2001-03-08 07.09.09                 DPSQLDAD SEL QUERY    105     8   273 0C2D7C4F00F4F4F2     0 SELECT * FROM
MXW2     MWSUMLD 2001-03-08 07.09.12                 DPSQLDAD SEL QUERY    105     8   273 0C2D7C4F00F4F4F2     0 SELECT * FROM
MXW2     MWSUMLD 2001-03-08 07.16.25                 DPSPURGD DELETE        58     8   273 0C2D7C4F00F4F4F2     0 DELETE FROM MX
MXW2     MWSUMLD 2001-03-08 07.17.04                 DPSPURGD DELETE        58    23   273 0C2D7C4F00F4F4F2     0 DELETE FROM MX
MXW2     MWSUMLD 2001-03-08 07.17.04                 DPSQLDAD SEL QUERY    105     8   273 0C2D7C4F00F4F4F2     0 SELECT * FROM
MXW2     MWSUMLD 2001-03-08 07.17.09                 DPSQLDAD SEL QUERY    105     8   273 0C2D7C4F00F4F4F2     0 SELECT * FROM
MXW2     MWSUMLD 2001-03-08 08.34.38                 DPSPURGD DELETE        58     8   273 0C2D7C4F00F4F4F2     0 DELETE FROM MX
MXW2     MWSUMLD 2001-03-08 08.35.55                 DPSPURGD DELETE        58    23   273 0C2D7C4F00F4F4F2     0 DELETE FROM MX
MXW2     MWSUMLD 2001-03-08 08.35.56                 DPSQLDAD SEL QUERY    105     8   273 0C2D7C4F00F4F4F2     0 SELECT * FROM
MXW2     MWSUMLD 2001-03-08 08.36.01                 DPSQLDAD SEL QUERY    105     8   273 0C2D7C4F00F4F4F2     0 SELECT * FROM
MXW2     MWSUMLD 2001-03-08 10.50.26                 DPSPURGD DELETE        58     8   273 0C2D7C4F00F4F4F2     0 DELETE FROM MX
MXW2     MWSUMLD 2001-03-08 10.50.31                 DPSPURGD DELETE        58    23   273 0C2D7C4F00F4F4F2     0 DELETE FROM MX
MXW2     MWSUMLD 2001-03-08 10.50.32                 DPSQLDAD SEL QUERY    105     8   273 0C2D7C4F00F4F4F2     0 SELECT * FROM
MXW2     MWSUMLD 2001-03-08 10.50.36                 DPSQLDAD SEL QUERY    105     8   273 0C2D7C4F00F4F4F2     0 SELECT * FROM
MXW2     MWSUMLD 2001-03-09 09.34.16                 DPSQLDAD SEL QUERY    120    81   273 0C2D7C4F00F0F0F0     0 SELECT * FROM
MXW2     RXDB2    2001-03-07 13.39.00 RXDB2P          RXSEL1M SEL QUERY    115    51   273 0C2D7C4F00F4F1F3     0 SELECT * FROM
MXW2     RXDB2    2001-03-07 13.46.43 RXDB2P          RXSEL1M SEL QUERY    115    51   273 0C2D7C4F00F4F1F3     0 SELECT * FROM
MXW2     RXDB2    2001-03-07 13.58.29 RXDB2P          RXSEL1M DELETE       477    51   273 0C2D7C4F00F4F1F3     0 DELETE FROM PL
MXW2     RXDB2    2001-03-07 13.58.35 RXDB2P          RXSEL1M EXPLAIN      477    38   273 0C2D7C4F00F4F1F3     0 EXPLAIN ALL SE
MXW2     RXDB2    2001-03-07 13.58.37 RXDB2P          RXSEL1M SEL QUERY    115    51   273 0C2D7C4F00F4F1F3     0 SELECT * FROM
MXW3     MWSUMLD 2001-03-07 17.38.25                 DPSPURGD DELETE        58     8   273 0C2D7C4F00F4F4F2     0 DELETE FROM MX
MXW3     MWSUMLD 2001-03-07 17.38.26                 DPSPURGD DELETE        58    23   273 0C2D7C4F00F4F4F2     0 DELETE FROM MX
MXW3     MWSUMLD 2001-03-07 17.38.26                 DPSQLDAD SEL QUERY    105     8   273 0C2D7C4F00F4F4F2     0 SELECT * FROM
MXW3     MWSUMLD 2001-03-07 17.38.36                 DPSQLDAD SEL QUERY    105     8   273 0C2D7C4F00F4F4F2     0 SELECT * FROM
MXW3     MWSUMLD 2001-03-08 11.28.51                 DPSPURGD DELETE        58     8   273 0C2D7C4F00F4F4F2     0 DELETE FROM MX
MXW3     MWSUMLD 2001-03-08 11.29.47                 DPSPURGD DELETE        58    23   273 0C2D7C4F00F4F4F2     0 DELETE FROM MX
MXW3     MWSUMLD 2001-03-08 11.29.48                 DPSQLDAD SEL QUERY    105     8   273 0C2D7C4F00F4F4F2     0 SELECT * FROM
MXW3     MWSUMLD 2001-03-08 11.29.57                 DPSQLDAD SEL QUERY    105     8   273 0C2D7C4F00F4F4F2     0 SELECT * FROM
```

Figure 79. DB2 DML at BIND Report

The report data is sorted by both AUTHID and PLAN (default).

AUTHID
Authorization ID of the user accessing this DB2 system.

PLAN
Plan executed.

DATE
Date of the BIND against an audited DB2 table.

TIME
Time of the BIND against an audited DB2 table.

PACKAGE COLL-ID
Package collection ID. This field is blank if the program is not bound as a package and does not belong to any package collection.

PROGRAM / DBRM
Name of the program or database request module (DBRM) containing the DML statement causing the BIND. This field contains the DBRM name for static BIND and the program name for dynamic BIND.

STMT TYPE
Type of DML statement issued. Possible values are

- CLOSE
- CREATE VIEW
- DELETE
- DESCRIBE
- EXEC IMMED
- EXECUTE
- EXPLAIN
- FETCH
- INSERT
- LOCK
- OPEN
- PREPARE
- SEL-BQUERY
- SEL-QUERY
- UPDATE

SQL STMT NUMBER
SQL statement number in the program or DBRM.

TABLE OBID
Internal DB2 identification (OBID) of the audited DB2 table. This field may be entered more than once on the report.

DATABASE
DBID containing the audited DB2 table.

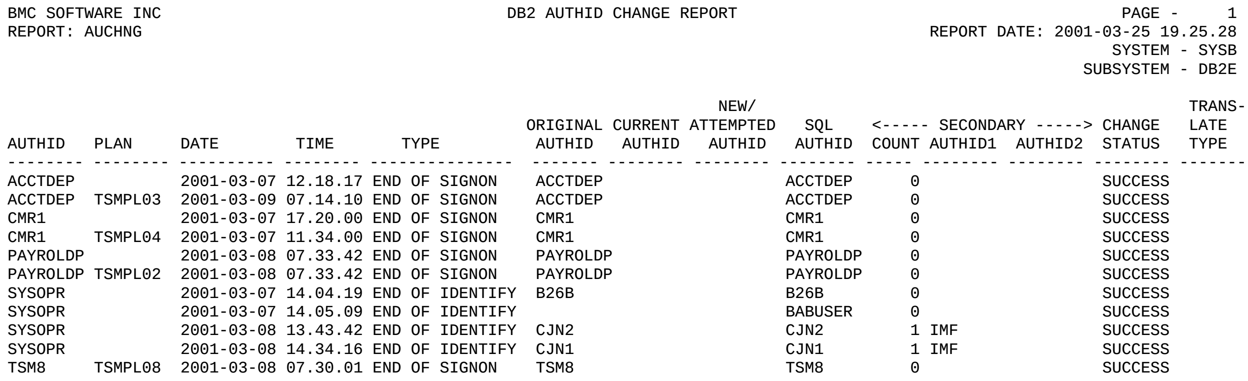
PRECOMPILE TIMESTAMP (HEX)
Hex dump of the DB2 timestamp of the program at precompile time.

RETURN CODE
SQL return code.

SQL TEXT
Beginning of the text of the SQL statement causing the BIND in the program or DBRM.

# AUCHNG—DB2 AUTHID Change Report

The DB2 AUTHID Change Report provides data on AUTHID change requests for each authorization ID and plan. AUTHIDs can be changed for a thread by a SET CURRENT SQLID statement, a CICS or IMS signon, an IMS, CICS, or TSO IDENTIFY request, or the translation of the AUTHID of a distributed request. It requires audit class 7 (TYPE2= AUDCHG).



```
BMC SOFTWARE INC                                    DB2 AUTHID CHANGE REPORT                                        PAGE -     1
REPORT: AUCHNG                                                                                  REPORT DATE: 2001-03-25 19.25.28
                                                                                                                   SYSTEM - SYSB
                                                                                                                SUBSYSTEM - DB2E

                                                                          NEW/                                             TRANS-
                                                      ORIGINAL CURRENT ATTEMPTED   SQL    <----- SECONDARY -----> CHANGE   LATE
AUTHID   PLAN     DATE        TIME       TYPE          AUTHID   AUTHID   AUTHID   AUTHID  COUNT AUTHID1  AUTHID2  STATUS   TYPE
-------- -------- ---------- -------- --------------   ------- -------- -------- -------- ----- -------- -------- -------- -------
ACCTDEP           2001-03-07 12.18.17 END OF SIGNON    ACCTDEP                   ACCTDEP      0                   SUCCESS
ACCTDEP  TSMPL03  2001-03-09 07.14.10 END OF SIGNON    ACCTDEP                   ACCTDEP      0                   SUCCESS
CMR1              2001-03-07 17.20.00 END OF SIGNON    CMR1                      CMR1         0                   SUCCESS
CMR1     TSMPL04  2001-03-07 11.34.00 END OF SIGNON    CMR1                      CMR1         0                   SUCCESS
PAYROLDP          2001-03-08 07.33.42 END OF SIGNON    PAYROLDP                  PAYROLDP     0                   SUCCESS
PAYROLDP TSMPL02  2001-03-08 07.33.42 END OF SIGNON    PAYROLDP                  PAYROLDP     0                   SUCCESS
SYSOPR            2001-03-07 14.04.19 END OF IDENTIFY  B26B                      B26B         0                   SUCCESS
SYSOPR            2001-03-07 14.05.09 END OF IDENTIFY                            BABUSER      0                   SUCCESS
SYSOPR            2001-03-08 13.43.42 END OF IDENTIFY  CJN2                      CJN2         1 IMF               SUCCESS
SYSOPR            2001-03-08 14.34.16 END OF IDENTIFY  CJN1                      CJN1         1 IMF               SUCCESS
TSM8     TSMPL08  2001-03-08 07.30.01 END OF SIGNON    TSM8                      TSM8         0                   SUCCESS
```

Figure 80. DB2 AUTHID Change Report

The report data is sorted by both AUTHID and PLAN (default).

AUTHID
: Authorization ID of the user accessing this DB2 system.

PLAN
: Plan name of the application program, transaction, or utility executed.

DATE
: Date of the authorization ID change.

TIME
: Time of the authorization ID change.

TYPE
: Type of authorization change or establishment. Possible values are

- SET CURRENT SQLID
- END OF IDENTIFY
- END OF SIGNON
- DISTR AUTH XLAT

ORIGINAL AUTHID
: Original value of the primary authorization ID passed to the IDENTIFY or SIGNON authorization exit (before any changes by the authorization exits).

CURRENT AUTHID
Initial value of the authorization ID of the user issuing the change request before the request is executed.

This column is valid only for SET CURRENT SQL and DISTR AUTH XLAT (distributed translation).

NEW / ATTEMPTED AUTHID
New value of the authorization ID of the user issuing the change request after the request is executed. For unsuccessful change requests, it is the value of the attempted change.

This column is valid only for SET CURRENT SQL and DISTR AUTH XLAT (distributed translation).

SQL AUTHID
Initial value of the SQL authorization ID from the IDENTIFY or SIGNON exit.

SECONDARY AUTHID(S)
Secondary authorization ID(s) set by the IDENTIFY or SIGNON exits. The number of secondary ID(s) and the first two values are shown.

CHANGE STATUS
Success or failure status of an authorization change request.

TRANSLATE TYPE
Inbound or outbound connection messages showing where the translation actually occurred for distributed requests.

# AUUTIL—DB2 Audit Utility Access Report

The DB2 Audit Utility Access Report provides utility data for each authorization ID and plan. It requires audit class 8 (TYPE2=AUDUTL).

```
BMC SOFTWARE INC                                   DB2 AUDIT UTILITY ACCESS REPORT                                   PAGE -      1
REPORT: AUTIL                                                                                     REPORT DATE: 2001-03-25 19.25.28
                                                                                                                     SYSTEM - SYSB
                                                                                                                  SUBSYSTEM - DB2E

                                                                                        PREV
                                        UTILITY         UTILITY  UTILITY     UTILITY    ITEM
 AUTHID    PLAN      DATE      TIME       ID             NAME     PHASE    ITEM TYPE    COUNT  DATABASE PAGESET
-------- -------- ---------- -------- ---------------- -------- -------- ------------- ------- -------- --------
MXW2     DSNUTIL  2001-03-07 12.33.47 DPSUMLD1         LOAD     UTILINIT                    0
MXW2     DSNUTIL  2001-03-07 14.41.11 DPSUMLD1         LOAD     UTILINIT                    0
MXW2     DSNUTIL  2001-03-08 07.09.18 DPSUMLD1         LOAD     UTILINIT                    0
MXW2     DSNUTIL  2001-03-08 07.10.05 DPSUMLD1         LOAD     RELOAD   PAGES              0 MWDMRDB1 DMRPRTAS
MXW2     DSNUTIL  2001-03-08 07.10.06 DPSUMLD1         LOAD     UTILTERM PAGES              0 MWDMRDB1 DMRPRTAS
MXW2     DSNUTIL  2001-03-08 07.17.15 DPSUMLD1         LOAD     UTILINIT                    0
MXW2     DSNUTIL  2001-03-08 07.17.57 DPSUMLD1         LOAD     UTILTERM PAGES              0 MWDMRDB1 DMRPRTAS
MXW2     DSNUTIL  2001-03-08 07.17.57 DPSUMLD1         LOAD     RELOAD   PAGES              0 MWDMRDB1 DMRPRTAS
MXW2     DSNUTIL  2001-03-08 08.36.10 DPSUMLD1         LOAD     UTILINIT                    0
MXW2     DSNUTIL  2001-03-08 08.36.57 DPSUMLD1         LOAD     RELOAD   PAGES              0 MWDMRDB1 DMRPRTAS
MXW2     DSNUTIL  2001-03-08 08.36.58 DPSUMLD1         LOAD     SORT     PAGES            156 MWDMRDB1 DMRPRTAS
MXW2     DSNUTIL  2001-03-08 08.37.01 DPSUMLD1         LOAD     BUILD    PAGES            156 MWDMRDB1 DMRPRTAS
MXW2     DSNUTIL  2001-03-08 08.37.04 DPSUMLD1         LOAD     REPORT   PAGES              8 MWDMRDB1 DMRPRTAS
MXW2     DSNUTIL  2001-03-08 08.37.04 DPSUMLD1         LOAD     INDEXVAL PAGES            156 MWDMRDB1 DMRPRTAS
MXW2     DSNUTIL  2001-03-08 08.37.06 DPSUMLD1         LOAD     UTILTERM PAGES              8 MWDMRDB1 DMRPRTAS
MXW2     DSNUTIL  2001-03-08 10.51.14 DPSUMLD1         LOAD     UTILINIT                    0
MXW2     DSNUTIL  2001-03-08 11.00.59 DPSUMLD1         LOAD     RELOAD   PAGES              0 MWDMRDB1 DMRPRTAS
MXW2     DSNUTIL  2001-03-08 11.02.47 DPSUMLD1         LOAD     UTILTERM PAGES              0 MWDMRDB1 DMRPRTAS
MXW3     DSNUTIL  2001-03-07 17.38.42 DPSUMLD1         LOAD     UTILINIT                    0
MXW3     DSNUTIL  2001-03-08 11.30.24 DPSUMLD1         LOAD     UTILINIT                    0
```

Figure 81.  DB2 Audit Utility Access Report

The report data is sorted by both AUTHID and PLAN (default).

AUTHID
Authorization ID of the user accessing this DB2 system.

PLAN
Plan name of the application program, transaction, or utility executed.

DATE
Date the utility accessed the DB2 object.

TIME
Time of the utility access against a DB2 table.

UTILITY ID
ID of the utility specified by the user for the utility run.

UTILITY NAME
Name of the executed utility.

UTILITY PHASE
ID of the utility phase.

UTILITY ITEM TYPE
Type of item involved in the current utility phase.

PREV ITEM COUNT
Number of items in the previous utility phase.

DATABASE
Name of the database (or DBID) containing the audited DB2 object.

PAGESET
Name of the page set (or OBID) containing the audited DB2 object.

## AUDTL—DB2 Audit Detail Report

The DB2 Audit Detail Report provides an overview of all audited activity per user. It requires audit classes 1 through 8 (TYPE2=AUDSUM).

```
BMC SOFTWARE INC                                   DB2 AUDIT DETAIL REPORT                                                PAGE -      1
REPORT: AUDTL                                                                                     REPORT DATE: 2001-03-25 19.25.28
                                                                                                                     SYSTEM - SYSB
                                                                                                                  SUBSYSTEM - DB2E

                                          AUDIT         AUDIT
 AUTHID    PLAN      DATE      TIME     CATEGORY        TYPE
-------- -------- ---------- -------- ------------- ----------------
ACCTDEP           2001-03-07 12.18.17 AUTH CHANGE   SIGNON
ACCTDEP  TSMPL03  2001-03-09 07.14.10 AUTH CHANGE   SIGNON
CJN1     DSNESPRR 2001-03-08 13.35.32 AUTH CNTL     GRANT
CJN2     QMF311   2001-03-08 13.50.22 AUTH FAIL     SELECT
CJN2     QMF311   2001-03-08 14.32.00 AUTH FAIL     INSERT
CMR1              2001-03-07 17.20.00 AUTH CHANGE   SIGNON
CMR1     TSMPL04  2001-03-07 11.34.00 AUTH CHANGE   SIGNON
ELC2     MXZAPPL  2001-03-08 17.26.16 AUTH CNTL     REVOKE
ELC2     MXZAPPL  2001-03-08 17.26.17 AUTH CNTL     GRANT
ELC2     MXZAPPL  2001-03-08 17.26.34 AUTH CNTL     REVOKE
ELC2     MXZAPPL  2001-03-08 17.26.42 AUTH CNTL     GRANT
LAA1     RXSEL1M  2001-03-07 13.17.05 AUTH FAIL     EXECUTE
MXW2     DSNUTIL  2001-03-07 12.33.47 UTILITY START LOAD
MXW2     DSNUTIL  2001-03-08 07.09.20 DML WRITES    FIRST ACCESS
MXW2     DSNUTIL  2001-03-08 07.10.05 UTILITY PHASE LOAD
MXW2     DSNUTIL  2001-03-08 07.10.06 UTILITY TERM  LOAD
MXW2     DSNUTIL  2001-03-08 08.37.04 UTILITY PHASE LOAD
MXW2     DSNUTIL  2001-03-08 08.37.04 DML READS     FIRST ACCESS
MXW2     MWSUMLD  2001-03-07 12.33.31 DML AT BIND   DELETE
MXW2     MWSUMLD  2001-03-07 12.33.37 DML AT BIND   SEL QUERY
MXW3     MWSUMLD  2001-03-08 11.30.01 DML READS     FIRST ACCESS
PAYROLDP          2001-03-08 07.33.42 AUTH CHANGE   SIGNON
PAYROLDP TSMPL02  2001-03-08 07.33.42 AUTH CHANGE   SIGNON
SYSOPR            2001-03-07 11.06.09 AUTH CHANGE   END OF IDENTIFY
SYSOPR            2001-03-09 11.01.24 AUTH FAIL     DISPLAY THD/DB
SYSOPR            2001-03-09 11.24.15 AUTH CHANGE   END OF IDENTIFY
SYSOPR            2001-03-09 11.25.18 AUTH FAIL     DISPLAY THD/DB
SYSOPR            2001-03-09 11.25.43 AUTH CHANGE   END OF IDENTIFY
TSM5              2001-03-07 15.56.03 AUTH CHANGE   SIGNON
TSM5     TSMPL05  2001-03-07 15.56.03 AUTH CHANGE   SIGNON
TSM6              2001-03-08 07.30.07 AUTH CHANGE   SIGNON
TSM6     TSMPL06  2001-03-08 07.30.07 AUTH CHANGE   SIGNON
TSM7              2001-03-07 16.48.38 AUTH CHANGE   SIGNON
TSM7     TSMPL07  2001-03-07 16.48.39 AUTH CHANGE   SIGNON
TSM8              2001-03-07 15.56.03 AUTH CHANGE   SIGNON
TSM8     TSMPL08  2001-03-09 07.42.43 AUTH CHANGE   SIGNON
```

Figure 82. DB2 Audit Detail Report

AUTHID
Authorization ID of the user accessing this DB2 system.

PLAN
Plan name of the application program, transaction, or utility executed.

DATE
Date the audit record completed.

TIME
Time the audit record completed.

AUDIT CATEGORY
Description of audit category, such as AUTH FAIL, DML READ, and so on.

AUDIT TYPE
Supplementary description of audited authorization.

# Part 4. Performance Data Tables

This part documents the Performance Reporter performance data tables so you can create queries and reports from these tables. The following performance data tables are described:

- The DB2 statistics tables (see Chapter 16, “DB2 Statistics Tables” on page 445)
- The DB2 detail and summary accounting tables (see Chapter 17, “DB2 Accounting Tables” on page 487)
- The DB2 audit tables (see Chapter 18, “DB2 Audit Tables” on page 529)

# Chapter 16. DB2 Statistics Tables

This chapter provides detailed information about each of the records in the Performance Reporter performance data tables you can use to produce statistics reports.

Statistics and buffer records are stored in three tables. These tables are DMRSTAT (Table 3), DDF (Table 4 on page 474), and DMRSBFDT (Table 5 on page 477). DMRSBFDT and DDF are optional.

## DMRSTAT—Statistics Table

One statistics record is created from each pair of SMF 100 records. This record is further processed to create a delta record, showing the changes in values during this statistics interval. Each row in the DMRSTAT table represents information about one statistics interval within DB2.

**Note:** The columns in Table 3 are shown in the same sequence as the DMRSTAT table. All columns not used (NO in *Used in Report* column below) in the predefined reports are defined at the end of the actual DB2 table after the dummy column, DMRAUTOCUST. These columns can be deleted as a group as an AutoCustomization option, or individually before the table is created. Data sharing columns are defined after the dummy column, DMRACSHARE; they can be deleted as a group as an AutoCustomization option as well.

Table 3. DMRSTAT Columns

| Performance Reporter Column Name | SMF Field Name | Used in Report | Field Type | Description |
|---|---|---|---|---|
| SYSTEMID | SM100SID | YES | Char(4) | System ID (SMF ID) |
| SUBSYSTEM | SM100SSI | YES | Char(4) | Subsystem ID - name of DB2 subsystem |
| DATETIME | QWHSSTCK | YES | Timestamp | Date and time record was created |
| DATE | QWHSSTCK | YES | Date | Date record was created |
| YEAR | QWHSSTCK | YES | Char(4) | Year record was created |
| MONTH | QWHSSTCK | YES | Char(2) | Month record was created |
| DAY | QWHSSTCK | YES | Char(2) | Day record was created |
| TIME | QWHSSTCK | YES | Time | Time record was created |
| HOUR | QWHSSTCK | YES | Char(2) | Hour record was created |
| DAYOFWEEK# | QWHSSTCK | NO | Smallint | Relative day of week, 1 to 7, where Monday=1 and Sunday=7 |
| DAYOFWEEK | QWHSSTCK | NO | Char(3) | MON, TUE, WED, THU, FRI, SAT, SUN |
| WEEK# | QWHSSTCK | NO | Integer | Relative week number |

Table 3. DMRSTAT Columns (Continued)

| Performance Reporter Column Name | SMF Field Name | Used in Report | Field Type | Description |
|---|---|---|---|---|
| LOCATION | QWHSLOCN | YES | Char(16) | Local location name (DB2 subsystem ID if not defined) |
| GROUPNAME | QWHADSGN | YES | Char(8) | Data sharing group name |
| MEMBERNAME | QWHAMEMN | YES | Char(8) | Data sharing member name |
| TRACEMASK | QWHSMTN | NO | Integer | Active trace mask |
| SSERVETCB | QWSAEJST of MSTR | YES | Decimal(15,6) | Job step CPU for MSTR |
| SSERVESRB | QWSASRBT of MSTR | YES | Decimal(15,6) | SRB CPU for MSTR |
| DBMANETCB | QWSAEJST of DBM | YES | Decimal(15,6) | Job step CPU for DBM |
| DBMANESRB | QWSASRBT of DBM | YES | Decimal(15,6) | SRB CPU for DBM |
| IRLMETCB | QWSAEJST of IRLM | YES | Decimal(15,6) | Job step CPU for IRLM |
| IRLMESRB | QWSASRBT of IRLM | YES | Decimal(15,6) | SRB CPU for IRLM |
| DISTTCB | QWSAEJST of DIST | YES | Decimal(15,6) | Job step TCB CPU for DIST address space |
| DISTSRB | QWSASRBT of DIST | YES | Decimal(15,6) | SRB CPU for DIST address space |
| SPASTCB | QWSAEJST of SPAS | YES | Decimal(15,6) | Job step TCB CPU for SPAS address space |
| SPASSRB | QWSASRBT of SPAS | YES | Decimal(15,6) | SRB CPU for SPAS address space |
| CREATETHREADS | Q3STCTHD | YES | Integer | Create threads |
| RIURREQ | Q3STRIUR | YES | Integer | Resolves |
| P1COMMITS | Q3STPREP | YES | Integer | Prepare to commit |
| P2COMMITS | Q3STCOMM | YES | Integer | Commits |
| ABORTS | Q3STABRT | YES | Integer | Rollbacks |
| SYNCCOMMITS | Q3STSYNC | YES | Integer | Synchronous commits |
| INDOUBT | Q3STINDT | YES | Integer | Indoubts |
| ENDOFTASKS | Q3STMEOT | YES | Integer | End of tasks |
| ENDOFMEMS | Q3STMEOM | YES | Integer | End of memories |
| CTHREADWAIT | Q3STCTHW | YES | Integer | Create threads that waited |

Table 3. DMRSTAT Columns (Continued)

| Performance Reporter Column Name | SMF Field Name | Used in Report | Field Type | Description |
|---|---|---|---|---|
| CHECKPOINTS | QWSDCKPT | YES | Integer | DB2 checkpoints |
| WRITENOWAIT | QJSTWRNW | YES | Integer | Log write nowait |
| WRITEFORCE | QJSTWRF | YES | Integer | Log write force |
| UABUFFWAIT | QJSTWTB | YES | Integer | Log write unavailable buffer |
| OBREADS | QJSTRBUF | YES | Integer | Log reads from buffer |
| ACTLREADS | QJSTRACT | YES | Integer | Log reads from active |
| ARCLREADS | QJSTRARH | YES | Integer | Log reads from archive |
| VOLCONTDELAY | QJSTTVC | YES | Integer | Number of read accesses delayed because of tape volume contention |
| BSDSREQ | QJSTBSDS | YES | Integer | BSDS requests |
| ALOGCIC | QJSTBFFL | YES | Integer | Active log output CIs |
| LWRCALLS | QJSTBFWR | YES | Integer | Log write requests |
| ARCLOGRALLOC | QJSTALR | YES | Integer | Archive log read allocations |
| ARCLOGWALLOC | QJSTALW | YES | Integer | Archive log write allocations |
| CNTLINTOFF | QJSTCIOF | YES | Integer | Number of control intervals offloaded |
| UNAVAILRESCS | QJSTWUR | YES | Integer | Number of read accesses delayed because of unavailable resources |
| LOOKAHEADATT | QJSTLAMA | YES | Integer | Number of look-ahead tape mounts attempted |
| LOOKAHEADSUC | QJSTLAMS | YES | Integer | Number of successful look-ahead tape mounts |
| LOGSUSPENDS | QJSTLSUS | YES | Integer | Number of times that a log manager request results in a suspend for a log record that is being written out to the log data sets |
| LOGWRITES | QJSTLOGW | YES | Integer | Total number of log write I/O requests (Media Manager calls) |
| LOGCIWRITES | QJSTCIWR | YES | Integer | Total number of log CIs written |
| LOGSERIALW | QJSTSERW | YES | Integer | Number of serial log write I/O requests |
| LOGTHRESHOLD | QJSTTHRW | YES | Integer | Number of times that an asynchronous log write request was scheduled because the log write threshold was reached |

Table 3. DMRSTAT Columns (Continued)

| Performance Reporter Column Name | SMF Field Name | Used in Report | Field Type | Description |
|---|---|---|---|---|
| LOGBUFPAGEIN | QJSTBPAG | YES | Integer | Number of times that a log output buffer was paged in before it could be initialized |
| CONVSDEALLOC | QDSTQCRT | YES | Integer | Number of conversations deallocated because ZPARM limit was reached for maximum concurrent remote threads |
| SELECTS | QXSELECT | YES | Integer | Number of selects |
| INSERTS | QXINSRT | YES | Integer | Number of inserts |
| UPDATES | QXUPDTE | YES | Integer | Number of updates |
| DELETES | QXDELET | YES | Integer | Number of deletes |
| DESCRIBES | QXDESC | YES | Integer | Number of describes |
| PREPARES | QXPREP | YES | Integer | Number of prepares |
| OPENS | QXOPEN | YES | Integer | Number of opens |
| FETCHES | QXFETCH | YES | Integer | Number of fetches |
| CLOSES | QXCLOSE | YES | Integer | Number of closes |
| CREATETBL | QXCRTAB | YES | Integer | Number of create table |
| CREATEINDX | QXCRINX | YES | Integer | Number of create index |
| CREATETSP | QXCTABS | YES | Integer | Number of create tablespace |
| CREATESYN | QXCRSYN | YES | Integer | Number of create synonym |
| CREATEDB | QXCRDAB | YES | Integer | Number of create database |
| CREATESG | QXCRSTG | YES | Integer | Number of create storage group |
| CREATEVU | QXDEFVU | YES | Integer | Number of create view |
| DROPINDX | QXDRPIX | YES | Integer | Number of drop index |
| DROPTBL | QXDRPTA | YES | Integer | Number of drop table |
| DROPTSP | QXDRPTS | YES | Integer | Number of drop tablespace |
| DROPDB | QXDRPDB | YES | Integer | Number of drop database |
| DROPSYN | QXDRPSY | YES | Integer | Number of drop synonym |
| DROPSG | QXDRPST | YES | Integer | Number of drop storage group |
| DROPVU | QXDRPVU | YES | Integer | Number of drop view |
| ALTERSG | QXALTST | YES | Integer | Number of alter storage group |
| ALTERTSP | QXALTTS | YES | Integer | Number of alter tablespace |

Table 3. DMRSTAT Columns (Continued)

| Performance Reporter Column Name | SMF Field Name | Used in Report | Field Type | Description |
|---|---|---|---|---|
| ALTERTBL | QXALTTA | YES | Integer | Number of alter table |
| ALTERINDX | QXALTIX | YES | Integer | Number of alter index |
| COMMENTON | QXCMTON | YES | Integer | Number of comment on |
| LOCKTBL | QXLOCK | YES | Integer | Number of lock table |
| GRANTS | QXGRANT | YES | Integer | Number of grants |
| REVOKES | QXREVOK | YES | Integer | Number of revokes |
| INCRBINDS | QXINCRB | YES | Integer | Number of incremental binds |
| LABELON | QXLABON | YES | Integer | Number of label on |
| SETSQLID | QXSETSQL | YES | Integer | Number of set current SQLID |
| MAXPARALLEL | QXMAXDEG | YES | Integer | Maximum degree of parallel I/O processing executed among all parallel groups |
| PARAGPSEXEC | QXTOTGRP | YES | Integer | Number of parallel groups executed |
| FALBAKCURSOR | QXDEGCUR | YES | Integer | Total number of parallel groups that fall back to sequential operation because of a cursor that can be used for update or delete |
| FALBAKESASOR | QXDEGESA | YES | Integer | Total number of parallel groups that fall back to sequential operation because of a lack of ESA sort support |
| FALBAKSTORBP | QXDEGBUF | YES | Integer | Total number of parallel groups that fall back to sequential operation because of storage shortage or contention on the buffer pool |
| PARAGPSLESS | QXREDGRP | YES | Integer | Total number of parallel groups processed to a parallel degree less than planned because of a storage shortage or contention on the buffer pool |
| PARAGPSPLAN | QXNORGRP | YES | Integer | Total number of parallel groups executed to the planned parallel degree |
| CONNECTTYPE1 | QXCON1 | YES | Integer | Number of CONNECT TYPE 1 statements executed |
| CONNECTTYPE2 | QXCON2 | YES | Integer | Number of CONNECT TYPE 2 statements executed |

Table 3. DMRSTAT Columns (Continued)

| Performance Reporter Column Name | SMF Field Name | Used in Report | Field Type | Description |
|---|---|---|---|---|
| RELEASESTMT | QXREL | YES | Integer | Number of RELEASE statements executed |
| SETCONNECT | QXSETCON | YES | Integer | Number of SET CONNECTION statements executed |
| SETCURRENTDG | QXSETCDG | YES | Integer | Number of SET CURRENT DEGREE statements executed |
| AAUTOBINDS | QTABINDA | YES | Integer | Autobind attempts |
| SAUTOBINDS | QTABIND | YES | Integer | Successful autobinds |
| AUTHCHECKS | QTAUCHK | YES | Integer | Authorization checks |
| SAUTHCHECKS | QTAUSUC | YES | Integer | Successful authorization checks |
| DEADLOCKS | QTXADEA | YES | Integer | Deadlocks |
| SUSPLOCK | QTXASLOC | YES | Integer | Suspends - lock conflict |
| TIMEOUTS | QTXATIM | YES | Integer | Timeouts |
| LOCKESHR | QTXALES | YES | Integer | Lock escalation to shared |
| LOCKEXCL | QTXALEX | YES | Integer | Lock escalation to exclusive |
| MAXPGLOCKS | QTXANPL | YES | Integer | Maximum number of page or row locks held |
| SUSPLATCH | QTXASLAT | YES | Integer | Suspend for latch |
| SUSPOTHER | QTXASOTH | YES | Integer | Suspend other reasons |
| LOCKREQS | QTXALOCK | YES | Integer | Lock requests |
| CLAIMREQUEST | QTXACLNO | YES | Integer | Number of claim requests |
| CLAIMREQUNSC | QTXACLUN | YES | Integer | Number of unsuccessful claim requests |
| DRAINREQUEST | QTXADRNO | YES | Integer | Number of drain requests |
| DRAINREQUNSC | QTXADRUN | YES | Integer | Number of unsuccessful drain requests |
| EDMFAILS | QISEFAIL | YES | Integer | EDM pool full failures |
| EDMPAGES | QISEPAGE | YES | Integer | Pages in EDM pool |
| CTREQS | QISECTG | YES | Integer | Requests for CT |
| LOADCTS | QISECTL | YES | Integer | Load CT sections |
| CTPAGES | QISECT | YES | Integer | Number pages for CT |
| FREEPAGES | QISEFREE | YES | Integer | Number of free pages |
| DBDPAGES | QISEDBD | YES | Integer | Number of pages for DBD |

Table 3. DMRSTAT Columns (Continued)

| Performance Reporter Column Name | SMF Field Name | Used in Report | Field Type | Description |
|---|---|---|---|---|
| SKCTPAGES | QISESKCT | YES | Integer | Number of pages for SKCT |
| DBDREQS | QISEDBDG | YES | Integer | Requests for DBD |
| DBDLOADS | QISEDBDL | YES | Integer | Load DBD |
| PKGREQS | QISEKTG | YES | Integer | Package requests |
| PKGLOAD | QISEKTL | YES | Integer | Package loads |
| PKGPAGE | QISEKT | YES | Integer | Pages for package table |
| SKPTPAGES | QISESKPT | YES | Integer | Skeleton package table pages |
| DYNSTMTINS | QISEDSI | YES | Integer | Number of inserts into the dynamic statement cache (DB2 5.1 and later only) |
| DYNSTMTREQ | QISEDSG | YES | Integer | Number of requests to the dynamic statement cache (DB2 5.1 and later only) |
| DYNSTMTPGS | QISEDSC | YES | Integer | Number of pages used for the dynamic statement cache (DB2 5.1 and later only) |
| DSPACEEDM | QISEDPGE | YES | Integer | Number of pages in the data space used by the EDM pool (DB2 6.1 and later only) |
| DSPACEFREE | QISEDFRE | YES | Integer | Number of free pages in the data space free chain (DB2 6.1 and later only) |
| DSPACEFULL | QISEDFAL | YES | Integer | Number of failures because the data space is full (DB2 6.1 and later only) |
| RIDTRMRDS | QISTRLLM | YES | Integer | RID terminated - over RDS limit |
| RIDTRMDM | QISTRPLM | YES | Integer | RID terminated - over DM limit |
| RIDHIBLKS | QISTRHIG | YES | Integer | RID high blocks allocated |
| RIDCURBLKS | QISTRCUR | YES | Integer | RID current blocks allocated |
| RIDTRMSTG | QISTRSTG | YES | Integer | RID terminated - no storage |
| RIDTRMNUM | QISTRMAX | YES | Integer | RID terminated - too many concurrent |
| COLSINVALIDS | QISTCOLS | YES | Integer | Number of COLUMNS INVALID SELECT encountered |
| DTHDQUED | QDSTQDBT | YES | Integer | DBAT queued count, because maximum number of active remote threads was reached |

Table 3. DMRSTAT Columns (Continued)

| Performance Reporter Column Name | SMF Field Name | Used in Report | Field Type | Description |
|---|---|---|---|---|
| SETCURRULES | QXSETCRL | YES | Integer | Set current rules statements executed |
| SQLCALL | QXCALL | YES | Integer | SQL call statements executed |
| SQLCALLAB | QXCALLAB | YES | Integer | Stored procedure abnormal executions |
| SQLCALLTO | QXCALLTO | YES | Integer | Stored procedure time outs |
| SQLCALLRJ | QXCALLRJ | YES | Integer | SQL call statements rejected |
| FALBAKENC | QXDEGENC | YES | Integer | Fallback to sequential because MVS/ESA enclave services not available |
| PARACOORNO | QXCOORNO | YES | Integer | Number of parallel groups executed on a single DB2 because of Coordinator value set to NO (DB2 5.1 and later only) |
| PARAISORR | QXISORR | YES | Integer | Total number of parallel groups executed on a single DB2 because the plan or package was bound with an isolation value of repeatable read (DB2 5.1 and later only) |
| CREATEGTT | QXCRGTT | YES | Integer | Number of CREATE GLOBAL TEMPORARY TABLE statements (DB2 5.1 and later only) |
| PARAXDSGRP | QXXCBPNX | YES | Integer | Number of parallel groups DB2 intended to run across the data sharing group (DB2 5.1 and later only) |
| PARACSKIP | QXXCSKIP | YES | Integer | Number of times the parallelism coordinator had to bypass a DB2 because of not enough buffer pool storage (DB2 5.1 and later only) |
| ASSOCLOCR | QXALOCL | YES | Integer | Number of associate locator statements executed (DB2 5.1 and later only) |
| ALLOCCUR | QXALOCC | YES | Integer | Number of allocate cursor statements executed (DB2 5.1 and later only) |
| PREPFND | QXSTFND | YES | Integer | Number of times DB2 satisfied a PREPARE request by making a copy of a statement in the prepared statement cache (DB2 5.1 and later only) |

Table 3. DMRSTAT Columns (Continued)

| Performance Reporter Column Name | SMF Field Name | Used in Report | Field Type | Description |
|---|---|---|---|---|
| PREPNOTFND | QXSTNFND | YES | Integer | Number of times DB2 searched the prepared statement cache but could not find a suitable prepared statement (DB2 5.1 and later only) |
| PREPIMPLICIT | QXSTIPRP | YES | Integer | Number of times DB2 did an implicit PREPARE for a statement bound with KEEPDYNAMIC(YES) because the prepared statement cache did not contain a valid copy of the prepared statement (DB2 5.1 and later only) |
| PREPNOIMPLICIT | QXSTNPRP | YES | Integer | Number of times DB2 did not PREPARE a statement bound with KEEPDYNAMIC(YES) because the prepared statement cache contained a valid copy of the prepared statement (DB2 5.1 and later only) |
| PREPDISCMAX | QXSTDEXP | YES | Integer | Number of times DB2 discarded a prepared statement from the prepared statement cache because MAXKEEPD was exceeded (DB2 5.1 and later only) |
| PREPDISCPROG | QXSTDINV | YES | Integer | Number of times DB2 discarded a prepared statement from the prepared statement cache because a program executed a DROP, ALTER, or REVOKE statement against a dependent object (DB2 5.1 and later only) |
| RENAMETBL | QXRNTAB | YES | Integer | Number of RENAME TABLE statements (DB2 5.1 and later only) |
| CREATETRIGGER | QXCTRIG | NO | Integer | Number of SQL CREATE TRIGGER statements (DB2 6.1 and later only) |
| DROPTRIGGER | QXDRPTR | NO | Integer | Number of SQL DROP TRIGGER statements (DB2 6.1 and later only) |
| SETCURRPATH | QXSETPTH | NO | Integer | Number of SQL SET CURRENT PATH statements (DB2 6.1 and later only) |
| DROPUDF | QXDRPFN | NO | Integer | Number of DROP UDF statements (DB2 6.1 and later only) |
| DROPPROC | QXDRPPR | NO | Integer | Number of DROP PROCEDURE statements (DB2 6.1 and later only) |

Table 3. DMRSTAT Columns (Continued)

| Performance Reporter Column Name | SMF Field Name | Used in Report | Field Type | Description |
|---|---|---|---|---|
| CREATEDISTINCT | QXCDIST | NO | Integer | Number of CREATE DISTINCT TYPE statements (DB2 6.1 and later only) |
| DROPDISTINCT | QXDDIST | NO | Integer | Number of DROP DISTINCT TYPE statements (DB2 6.1 and later only) |
| CREATEFUNC | QXCRUDF | NO | Integer | Number of CREATE FUNCTION statements (DB2 6.1 and later only) |
| CREATEPROC | QXCRPRO | NO | Integer | Number of CREATE PROCEDURE statements (DB2 6.1 and later only) |
| HOLDLOCATOR | QXHOLDL | NO | Integer | Number of HOLD LOCATOR statements (DB2 6.1 and later only) |
| FREELOCATOR | QXFREEL | NO | Integer | Number of FREE LOCATOR statements (DB2 6.1 and later only) |
| RACSUCC | QTRACAUT | NO | Integer | Number of successful authorization checks for user-defined functions or stored procedures that use the routine authorization cache, including public authorization checks (QTRACPUB) (DB2 6.1 and later only) |
| RACPUBLIC | QTRACPUB | NO | Integer | Number of successful authorization checks for user-defined function or stored procedure execution authority when that authority is held by public (DB2 6.1 and later only) |
| RACNOTUSED | QTRACNOT | NO | Integer | Number of authorization checks for user-defined function or stored procedure execution authority that could not use the routine authorization cache (DB2 6.1 and later only) |
| RACOWID | QTRACOW1 | NO | Integer | Number of times that DB2 wrote over an authorization ID in the routine authorization cache (DB2 6.1 and later only) |
| RACOWENT | QTRACOW2 | NO | Integer | Number of times that DB2 wrote over a routine entry in the routine authorization cache (DB2 6.1 and later only) |

Table 3. DMRSTAT Columns (Continued)

| **Performance Reporter Column Name** | **SMF Field Name** | **Used in Report** | **Field Type** | **Description** |
|---|---|---|---|---|
| RACNOTADD | QTRACNAC | NO | Integer | Number of times that DB2 could not add an entry to the routine authorization cache (DB2 6.1 and later only) |
| PARACONFIG | QXREPOP1 | NO | Integer | Number of parallel groups for which DB2 reformulated the parallel portion of the access path because the SYSPLEX configuration at run time was different from the SYSPLEX configuration at bind time. This counter is incremented only by the parallelism coordinator at run time (DB2 6.1 and later only) |
| PARANOBP | QXREPOP2 | NO | Integer | Number of parallel groups for which DB2 reformulated the parallel portion of the access path because there was not enough buffer pool resource. This counter is incremented only by the parallelism coordinator at run time (DB2 6.1 and later only) |
| CREATEAUXTBL | QXCRATB | NO | Integer | Number of CREATE AUXILIARY TABLE statements (DB2 6.1 and later only) |
| MAXLOBSTG | QXSTLOBV | NO | Integer | Maximum storage used for LOB values, in megabytes (DB2 6.1 and later only) |
| ALTERFUNC | QXALUDF | NO | Integer | Number of ALTER FUNCTION statements (DB2 6.1 and later only) |
| ALTERPROC | QXALPRO | NO | Integer | Number of ALTER PROCEDURE statements (DB2 6.1 and later only) |
| DIRECTROW | QXROIMAT | NO | Integer | Number of times that DB2 used direct row access to locate a record (DB2 6.1 and later only) |
| DIRECTROWIX | QXROIIDX | NO | Integer | Number of times that DB2 attempted to use direct row access but reverted to using an index to locate a record (DB2 6.1 and later only) |
| DIRECTROWTS | QXROITS | NO | Integer | Number of times that DB2 attempted to use direct row access but reverted to using a table space scan to locate a record (DB2 6.1 and later only) |

Table 3. DMRSTAT Columns (Continued)

| **Performance Reporter Column Name** | **SMF Field Name** | **Used in Report** | **Field Type** | **Description** |
|---|---|---|---|---|
| STMTTRIGGER | QXSTTRG | YES | Integer | Number of times a statement trigger is activated (DB2 6.1 and later only) |
| ROWTRIGGER | QXROWTRG | YES | Integer | Number of times a row trigger is activated (DB2 6.1 and later only) |
| SQLERRTRIGGER | QXTRGERR | YES | Integer | Number of times an SQL error occurred during the execution of a triggered action (DB2 6.1 and later only) |
| MAXSQLCASCADE | QXCASCDP | NO | Integer | Maximum level of nested SQL cascading due to triggers, user-defined functions, and stored procedures (DB2 6.1 and later only) |
| FUNC | QXCAUD | YES | Integer | Number of user-defined functions executed (DB2 6.1 and later only) |
| FUNCAB | QXCAUDAB | YES | Integer | Number of times a user-defined function abended (DB2 6.1 and later only) |
| FUNCTO | QXCAUDTO | YES | Integer | Number of times a user-defined function timed out waiting to be scheduled (DB2 6.1 and later only) |
| FUNCRJ | QXCAUDRJ | YES | Integer | Number of times a user-defined function was rejected (DB2 6.1 and later only) |
| SETSYSPARM | Q9STCTX4 | NO | Integer | Set SYSPARM commands (DB2 7.1 and later only) |
| DBATNEEDED | QDSTNDBA | NO | Integer | Requests requiring DBATs (DB2 7.1 and later only) |
| DBATPOOL | QDSTPOOL | NO | Integer | Requests assigned POOL threads (DB2 7.1 and later only) |
| SETCURPREC | QXSETCPR | NO | Integer | SET CURRENT PRECISION statements (DB2 7.1 and later only) |
| DCLGLOBALTT | QXDCLGTT | NO | Integer | DECLARE GLOBAL TEMPORARY TABLE statements (DB2 7.1 and later only) |
| PARAGLOBALTT | QXDEGDTT | NO | Integer | Parallel groups using DECLARE TEMPORARY TABLE (DB2 6.1 and later only) |
| BPGETPAGE | QBSTGET | YES | Integer | Number of GETPAGES |
| BPREADS | QBSTRIO | YES | Integer | Synchronous read I/O |

Table 3. DMRSTAT Columns (Continued)

| Performance Reporter Column Name | SMF Field Name | Used in Report | Field Type | Description |
|---|---|---|---|---|
| BPDWV | QBSTDWV | YES | Integer | Number of times the vertical deferred write threshold was reached |
| BPFAILMAX | QBSTXFL | YES | Integer | Expansions failed - pool was full |
| BPFAILSOS | QBSTXFV | YES | Integer | Expansions failed - GETMAIN |
| BPPGUPDAT | QBSTSWS | YES | Integer | Number of page update requests |
| BPPGWRITN | QBSTPWS | YES | Integer | Pages written |
| BPASYNCWR | QBSTWIO | YES | Integer | Asynchronous writes |
| BPACTBUFF | QBSTCBA | YES | Integer | Current active buffers |
| BPPAGEINRD | QBSTRPI | YES | Integer | Pageins for read I/O |
| BPPAGEINWR | QBSTWPI | YES | Integer | Pageins for write I/O |
| BPOPENSOK | QBSTDSO | YES | Integer | Number of successful opens |
| BPSYNCWR | QBSTIMW | YES | Integer | Immediate (synchronous) write I/O |
| BPPFREQS | QBSTSEQ | YES | Integer | Number of sequential prefetch requests |
| BPFPAGESRD | QBSTSPP | YES | Integer | Prefetch pages read |
| BPPFNOBFR | QBSTSPD | YES | Integer | Prefetch disabled - no buffers |
| BPPFNORDN | QBSTREE | YES | Integer | Prefetch disabled - no read engine |
| BPWEUNAVAL | QBSTWEE | YES | Integer | Prefetch disabled - write engine |
| BPDEFWRITE | QBSTDWT | YES | Integer | Times deferred write threshold reached |
| BPDMCRITIC | QBSTDMC | YES | Integer | Times DM critical threshold reached |
| BPMIGDS | QBSTMIG | YES | Integer | Migrated data sets encountered |
| BPRTO | QBSTRTO | YES | Integer | Number of recall timeouts |
| BPPIO | QBSTPIO | YES | Integer | Number of asynchronous read I/Os because of sequential prefetch |
| BPPFNOWKF | QBSTWKPD | YES | Integer | Prefetch abort - zero quantity |
| BPMAX | QBSTMAX | YES | Integer | Number of work files not created because of insufficient buffer resources |
| BPMAXWKFIL | QBSTWFM | YES | Integer | Maximum work files in merge |
| BPDISTREAD | QBSTWDRP | YES | Integer | Pages for destructive read |

Table 3. DMRSTAT Columns (Continued)

| Performance Reporter Column Name | SMF Field Name | Used in Report | Field Type | Description |
|---|---|---|---|---|
| BPDEQVDWQD | QBSTWBVQ | YES | Integer | Dequeue from VDWQ for destructive read |
| BPMERGPASS | QBSTWFR | YES | Integer | Number of merge passes |
| BPWKFLMERG | QBSTWFT | YES | Integer | Total work files in merge |
| BPWKREJBUF | QBSTWFD | YES | Integer | Work files rejected / no buffer |
| BPMERGNBUF | QBSTWFF | YES | Integer | Merge passes with insufficient buffers |
| BPLISTPREF | QBSTLPF | YES | Integer | List prefetch requests |
| BPDPF | QBSTDPF | YES | Integer | Number of dynamic prefetch requests |
| BPVPA | QBSTVPA | YES | Integer | Number of successful virtual buffer pool expansions or contractions because of the ALTER BUFFERPOOL command |
| BPHPA | QBSTHPA | YES | Integer | Number of successful hiperpool expansions or contractions because of the ALTER BUFFERPOOL command |
| BPHRE | QBSTHRE | YES | Integer | Number of successful synchronous requests to move a page from a hiperpool to a virtual buffer pool |
| BPHRA | QBSTHRA | YES | Integer | Number of pages moved successfully from the hiperpool to the virtual buffer pool asynchronously |
| BPHRF | QBSTHRF | YES | Integer | Number of pages for which a synchronous or asynchronous read request failed because the backing expanded storage page was stolen by the system |
| BPHWR | QBSTHWR | YES | Integer | Number of successful requests issued by DB2 to synchronously move a page from the virtual buffer pool to the hiperpool |
| BPHWA | QBSTHWA | YES | Integer | Number of pages successfully moved asynchronously from the virtual buffer pool to the hiperpool |
| BPHWF | QBSTHWF | YES | Integer | Number of pages for which a synchronous or asynchronous write request failed because of a shortage of expanded storage |

Table 3. DMRSTAT Columns (Continued)

| Performance Reporter Column Name | SMF Field Name | Used in Report | Field Type | Description |
|---|---|---|---|---|
| BPVPL | QBSTVPL | YES | Integer | Number of buffer pools allocated for a virtual buffer pool |
| BPHPL | QBSTHPL | YES | Integer | Number of buffers allocated for a hiperpool |
| BPDPP | QBSTDPP | YES | Integer | Number of asynchronous page reads because of dynamic prefetch |
| BPLPP | QBSTLPP | YES | Integer | Number of asynchronous page reads because of list prefetch |
| BPDIO | QBSTDIO | YES | Integer | Number of asynchronous read I/Os because of dynamic prefetch |
| BPLIO | QBSTLIO | YES | Integer | Number of asynchronous read I/Os because of list prefetch |
| BPSGT | QBSTSGT | YES | Integer | Number of GETPAGE requests issued by sequential access requesters |
| BPSIO | QBSTSIO | YES | Integer | Number of synchronous read I/Os issued by sequential access requesters |
| BPNGT | QBSTNGT | YES | Integer | Number of times conditional GETPAGE requests could not be satisfied for this buffer pool |
| BPXIS | QBSTXIS | YES | Integer | Highest number of concurrent prefetch I/O streams allocated for supporting queries processed in parallel in this buffer pool |
| BPJIS | QBSTJIS | YES | Integer | Number of requested prefetch I/O streams denied because of a storage shortage in the buffer pool |
| BPPQO | QBSTPQO | YES | Integer | Number of requests made for processing queries in parallel in this buffer pool |
| BPPQF | QBSTPQF | YES | Integer | Number of times during this statistics interval DB2 could not allocate the requested number of buffer pages to allow a parallel group to run to the planned degree |
| BPPL1 | QBSTPL1 | YES | Integer | Number of occurrences when the prefetch quantity is reduced from normal to one-half of normal |

Table 3. DMRSTAT Columns (Continued)

| Performance Reporter Column Name | SMF Field Name | Used in Report | Field Type | Description |
|---|---|---|---|---|
| BPPL2 | QBSTPL2 | YES | Integer | Number of occurrences when the prefetch quantity is reduced from one-half to one-quarter of normal |
| BPARA | QBSTARA | YES | Integer | Number of pages moved successfully from the hiperpool to the virtual buffer pool by the asynchronous data mover facility |
| BPARF | QBSTARF | YES | Integer | Number of pages for which a read request, using the asynchronous data mover facility, failed because the backing expanded storage was stolen or some other error occurred |
| BPAWA | QBSTAWA | YES | Integer | Number of pages moved successfully from the virtual buffer pool to the hiperpool by the asynchronous data mover facility |
| BPAWF | QBSTAWF | YES | Integer | Number of pages for which a write request, using the asynchronous data mover facility, failed because the backing expanded storage was stolen or some other error occurred |
| BPHBE | QBSTHBE | YES | Integer | Number of hiperpool buffers currently backed by expanded storage |
| BPLPLADDS | QBSTLPL | NO | Integer | Number of times pages added to LPL (DB2 7.1 and later only) |
| DMRACSHARE | none | NO | Integer | DMR column used only as a marker during AutoCustomization to delete data sharing columns, if requested |
| GBPREADINVBD | QBGLXD | NO | Integer | Synchronous coupling facility reads caused by invalid buffer and with data returned |
| GBPREADINVBR | QBGLXR | NO | Integer | Synchronous coupling facility reads caused by invalid buffer with no data returned and a directory entry created |
| GBPREADINVBN | QBGLXN | NO | Integer | Synchronous coupling facility reads caused by invalid buffer with no data returned and no directory entry created |
| GBPREADNOPGD | QBGLMD | NO | Integer | Synchronous coupling facility reads caused by page not in buffer pool and with data returned |

Table 3. DMRSTAT Columns (Continued)

| Performance Reporter Column Name | SMF Field Name | Used in Report | Field Type | Description |
|---|---|---|---|---|
| GBPREADNOPGR | QBGLMR | NO | Integer | Synchronous coupling facility reads caused by page not in buffer pool with no data returned and a directory entry created |
| GBPREADNOPGN | QBGLMN | NO | Integer | Synchronous coupling facility reads caused by page not in buffer pool with no data returned and no directory entry created |
| GBPWRITCHG | QBGLSW | NO | Integer | Changed pages written synchronously to group buffer pool |
| GBPWRITCLEAN | QBGLWC | NO | Integer | Clean pages written synchronously to group buffer pool |
| GBPCASTCLASS | QBGLCT | NO | Integer | Group buffer pool castout because class threshold detected |
| GBPCASTGBP | QBGLGT | NO | Integer | Group buffer pool castout because group buffer pool threshold detected |
| GBPAREADD | QBGLAD | NO | Integer | Asynchronous coupling facility reads with data returned |
| GBPAREADR | QBGLAR | NO | Integer | Asynchronous coupling facility reads without data returned and directory |
| GBPAREADN | QBGLAN | NO | Integer | Asynchronous coupling facility reads without data returned and no directory |
| GBPAWRITCHG | QBGLAW | NO | Integer | Changed pages asynchronously written to group buffer pool |
| GBPAWRITCLEAN | QBGLAC | NO | Integer | Clean pages asynchronously written to group buffer pool |
| GBPCASTOPS | QBGLRC | NO | Integer | Pages cast out |
| GBPNOCASTENG | QBGLCN | NO | Integer | Castout engine not available ( |
| GBPNOWRTENG | QBGLSU | NO | Integer | Coupling facility write engine not available |
| GBPNOREAD | QBGLRF | NO | Integer | Coupling facility read not complete |
| GBPNOWRITE | QBGLWF | NO | Integer | Coupling facility write not complete |
| GBPOTHER | QBGLOS | NO | Integer | Other coupling facility requests (DB2 4.1 only) |
| GBPRDSTGST | QBGLOS | NO | Integer | Number of coupling facility requests to read storage statistics (DB2 5.1 and later only) |

Table 3. DMRSTAT Columns (Continued)

| **Performance Reporter Column Name** | **SMF Field Name** | **Used in Report** | **Field Type** | **Description** |
|---|---|---|---|---|
| GBPCHKPT | QBGLCK | NO | Integer | Number of group buffer pool checkpoints triggered by this member (DB2 5.1 and later only) |
| GBPREBUILD | QBGLRB | NO | Integer | Number of group buffer pool rebuilds in which this member participated (DB2 5.1 and later only) |
| GBPUNLKCAST | QBGLUN | NO | Integer | Number of coupling facility requests to unlock the castout lock on the pages (DB2 5.1 and later only) |
| GBPRDCASTCL | QBGLCC | NO | Integer | Number of coupling facility requests to read the castout class (DB2 5.1 and later only) |
| GBPRDCASTST | QBGLCS | NO | Integer | Number of coupling facility requests to read the castout statistics (DB2 5.1 and later only) |
| GBPDELETE | QBGLDN | NO | Integer | Number of group buffer pool requests to delete all directory and data entries for a page set or partition (DB2 5.1 and later only) |
| GBPRDDIR | QBGLRD | NO | Integer | Number of coupling facility requests to read directory information (DB2 5.1 and later only) |
| GBPREGPG | QBGLRG | NO | Integer | Number of coupling facility requests to register a page (DB2 5.1 and later only) |
| GBPUNREGPG | QBGLDG | NO | Integer | Number of coupling facility requests to unregister a page (DB2 5.1 and later only) |
| GBPREGPGLST | QBGLAX | NO | Integer | Number of requests to register a page list in the coupling facility |
| GBPRETVCHP | QBGLAY | NO | Integer | Number of coupling facility reads to retrieve a changed page from the group buffer pool as a result of feedback from the request to register a page list |
| GBPRETVCLP | QBGLAZ | NO | Integer | Number of coupling facility reads to retrieve a clean page from the group buffer pool as a result of feedback from the request to register a page list |

Table 3. DMRSTAT Columns (Continued)

| Performance Reporter Column Name | SMF Field Name | Used in Report | Field Type | Description |
|---|---|---|---|---|
| GBPEXPLICITXI | QBGLEX | YES | Integer | Number of explicit cross-invalidations (DB2 6.1 and later only) |
| GBPWRITCHG2 | QBGL2W | YES | Integer | Number of coupling facility requests to write changed pages to the secondary group buffer pool for duplexing (DB2 6.1 and later only) |
| GBPWRITSTG2 | QBGL2F | YES | Integer | Number of coupling facility requests to write changed pages to the secondary group buffer pool for duplexing that failed due to a lack of storage in the coupling facility (DB2 6.1 and later only) |
| GBPWRITCHK2 | QBGL2S | NO | Integer | Number of completion checks for writes to the secondary GBP that were suspended because the write had not yet completed (DB2 6.1 and later only) |
| GBPDELNMLST2 | QBGL2D | YES | Integer | Number of group buffer pool requests to the secondary group buffer pool to delete a list of pages after they have been cast out from the primary group buffer pool (DB2 6.1 and later only) |
| GBPRDCASTST2 | QBGL2R | YES | Integer | Number of coupling facility requests to read the castout statistics for the secondary group buffer pool. Issued by group buffer pool structure owner to check for orphaned data entries in the secondary (DB2 6.1 and later only) |
| GBPDELNM2 | QBGL2N | YES | Integer | Number of group buffer pool requests to delete a page from the secondary group buffer pool. Issued by group buffer pool structure owner to delete orphaned data entries in the secondary as part of the garbage collection logic (DB2 6.1 and later only) |
| GBPASYNPRIM | QBGLHS | NO | Integer | Asynchronous requests for primary GBP (DB2 7.1 and later only) |
| GBPASYNSEC | QBGL2H | NO | Integer | Asynchronous requests for secondary GBP (DB2 6.1 and later only) |

Table 3. DMRSTAT Columns (Continued)

| Performance Reporter Column Name | SMF Field Name | Used in Report | Field Type | Description |
|---|---|---|---|---|
| GBPDEPGETPG | QBGLGG | NO | Integer | Getpages for GBP-dependent pages (DB2 7.1 and later only) |
| GBPPLKSPMAP | QBGLP1 | NO | Integer | Page P-Lock requests for space map pages (DB2 7.1 and later only) |
| GBPPLKDATA | QBGLP2 | NO | Integer | Page P-Lock requests for data pages (DB2 7.1 and later only) |
| GBPPLKIDX | QBGLP3 | NO | Integer | Page P-Lock requests for index leaf pages (DB2 7.1 and later only) |
| GBPPLKUNLK | QBGLU1 | NO | Integer | Page P-Lock unlock requests (DB2 7.1 and later only) |
| GBPPSUSSPMAP | QBGLS1 | NO | Integer | Page P-Lock suspensions for space map pages (DB2 7.1 and later only) |
| GBPPSUSDATA | QBGLS2 | NO | Integer | Page P-Lock suspensions for data pages (DB2 7.1 and later only) |
| GBPPSUSIDX | QBGLS3 | NO | Integer | Page P-Lock suspensions for index leaf pages (DB2 7.1 and later only) |
| GBPPNEGSPMAP | QBGLN1 | NO | Integer | Page P-Lock negotiations for space map pages (DB2 7.1 and later only) |
| GBPPNEGDATA | QBGLN2 | NO | Integer | Page P-Lock negotiations for data pages (DB2 7.1 and later only) |
| GBPPNEGIDX | QBGLN3 | NO | Integer | Page P-Lock negotiations for index leaf pages (DB2 7.1 and later only) |
| GLPLOCKLK | QTGSLPLK | NO | Integer | Lock requests for p-locks |
| GLPLOCKCHG | QTGSCPLK | NO | Integer | Change requests for p-locks |
| GLPLOCKUNLK | QTGSUPLK | NO | Integer | Unlock requests for p-locks |
| GLXESSYNCLK | QTGSLSLM | NO | Integer | L-lock and p-lock lock requests propagated to MVS XES synchronously |
| GLXESSYNCCHG | QTGSCSLM | NO | Integer | L-lock and p-lock change requests propagated to MVS XES synchronously |
| GLXESSYNCUNLK | QTGSUSLM | NO | Integer | L-lock and p-lock unlock requests propagated to MVS XES synchronously |
| GLSUSPIRLM | QTGSIGLO | NO | Integer | Suspends caused by IRLM global resource contention |
| GLSUSPXES | QTGSSGLO | NO | Integer | Suspends caused by MVS XES global resource contention |

Table 3. DMRSTAT Columns (Continued)

| Performance Reporter Column Name | SMF Field Name | Used in Report | Field Type | Description |
|---|---|---|---|---|
| GLSUSPFALSE | QTGSFLSE | NO | Integer | Suspends caused by false contentions |
| GLINCOMPAT | QTGSDRTA | NO | Integer | Global lock or change requests denied for incompatible retained lock |
| GLNOTFYSENT | QTGSNTFY | NO | Integer | Notify messages sent |
| GLNOTFYRECVD | QTGSNTFR | NO | Integer | Notify messages received |
| GLIRLMXESASYN | QTGSKIDS | NO | Integer | Resources propagated by IRLM to MVS XES asynch |
| GLNEGPLOCKPP | QTGSPPPE | NO | Integer | Negotiate page set or partition because of changing inter-DB2 interest levels |
| GLNEGPLOCKPG | QTGSPGPE | NO | Integer | Negotiate page p-lock because of inter-DB2 contention |
| GLNEGPLOCKOT | QTGSOTPE | NO | Integer | Negotiate another p-lock type |
| GLNEGPLOCKCH | QTGSCHNP | NO | Integer | P-lock change requests during p-lock negotiation |
| GLEXITMAXENG | QTGSPEMX | NO | Integer | Maximum engines available for p-lock exit or notify exit requests |
| GLENGUNAVAIL | QTGSPEQW | NO | Integer | Engine not available for p-lock exit or notify exit request |
| SMFWRITEOK | QWSBSRSW of SMF | NO | Integer | SMF records written |
| SMFNOWRITE | QWSBSRNW of SMF | NO | Integer | SMF records not written |
| SMFBUFFERR | QWSBSBUF of SMF | NO | Smallint | SMF buffer errors |
| SMFNOTACTERR | QWSBSACT of SMF | NO | Smallint | SMF not active errors |
| SMFNOTACC | QWSBSRNA of SMF | NO | Smallint | SMF records not accepted |
| SMFWRITEFAILS | QWSBSWF of SMF | NO | Smallint | SMF writer failures |
| GTFWRITEOK | QWSBSRSW of GTF | NO | Integer | GTF records written |
| OTHWRITEOK | QWSBSRSW of other | NO | Integer | Other records written |

Table 3. DMRSTAT Columns (Continued)

| Performance Reporter Column Name | SMF Field Name | Used in Report | Field Type | Description |
|---|---|---|---|---|
| I12WRITEOK | QWSCSRSW of IFCID 1 or 2 | NO | Integer | IFCID 1 or 2 records written |
| I12NOWRITE | QWSCSRNW of IFCID 1 or 2 | NO | Integer | IFCID 1 or 2 records not written |
| I3WRITEOK | QWSCSRSW of IFCID 3 | NO | Integer | IFCID 3 records written |
| I3NOWRITE | QWSCSRNW of IFCID 3 | NO | Integer | IFCID 3 records not written |
| I4WRITEOK | QWSCSRSW of other IFCIDs | NO | Integer | Other IFCIDs written |
| I4NOWRITE | QWSCSRNW of other IFCIDs | NO | Integer | Other IFCIDs not written |
| IDENTIFIES | Q3STIDEN | NO | Integer | Identify requests |
| SIGNONS | Q3STSIGN | NO | Integer | Signons |
| TERMINATES | Q3STTERM | NO | Integer | Terminates |
| EXITREQ | Q3STEXIT | NO | Integer | Exit requests |
| SSICALLS | Q3STSSSI | NO | Integer | Number of SSI calls processed |
| ROCOMMITS | Q3STRDON | NO | Integer | Read-only commits |
| DISDB | Q9STCTR0 | NO | Integer | Display database commands |
| DISTHREAD | Q9STCTR1 | NO | Integer | Display thread commands |
| DISUTIL | Q9STCTR2 | NO | Integer | Display utility commands |
| RECOVERBSDS | Q9STCTR3 | NO | Integer | Recover BSDS commands |
| RECOVERINDB | Q9STCTR4 | NO | Integer | Recover indoubt commands |
| STARTDB | Q9STCTR5 | NO | Integer | Start database commands |
| STARTTRC | Q9STCTR6 | NO | Integer | Start trace commands |
| STARTDB2 | Q9STCTR7 | NO | Integer | Start DB2 commands |
| STOPDB | Q9STCTR8 | NO | Integer | Stop database commands |
| STOPTRC | Q9STCTR9 | NO | Integer | Stop trace commands |
| STOPDB2 | Q9STCTRA | NO | Integer | Stop DB2 commands |
| TERMUTIL | Q9STCTRB | NO | Integer | Terminate utility commands |
| DISTRC | Q9STCTRC | NO | Integer | Display trace commands |
| RESETGENLU | Q9STCTRD | NO | Integer | Reset genericlu commands |
| ERRORS | Q9STEROR | NO | Integer | Unrecognized commands |

Table 3. DMRSTAT Columns (Continued)

| Performance Reporter Column Name | SMF Field Name | Used in Report | Field Type | Description |
|---|---|---|---|---|
| STARTRLIMIT | Q9STCTRE | NO | Integer | Start rlimit commands |
| STOPRLIMIT | Q9STCTRF | NO | Integer | Stop rlimit commands |
| DISRLIMIT | Q9STCTRG | NO | Integer | Display rlimit commands |
| MODTRACE | Q9STCTRH | NO | Integer | Modify trace commands |
| STARTDDF | Q9STCTRI | NO | Integer | Start DDF commands |
| STOPDDF | Q9STCTRJ | NO | Integer | Stop DDF commands |
| CANCELDDFTHD | Q9STCTRK | NO | Integer | Cancel DDF thread commands |
| DISLOCATION | Q9STCTRL | NO | Integer | Display location commands |
| ARCHLOG | Q9STCTRM | NO | Integer | Archive log commands |
| SETARCHIVE | Q9STCTRP | NO | Integer | Number of SET ARCHIVE commands |
| DISPLAYARCH | Q9STCTRQ | NO | Integer | Number of DISPLAY ARCHIVE commands |
| ALTERBP | Q9STCTRN | NO | Integer | Number of ALTER BUFFERPOOL commands |
| DISPLAYBP | Q9STCTRO | NO | Integer | Number of DISPLAY BUFFERPOOL commands |
| RESETINDOUBT | Q9STCTRR | NO | Integer | Number of RESET INDOUBT commands |
| ALTERGBP | Q9STCTRS | NO | Integer | Alter groupbufferpool commands |
| DISPLAYGBP | Q9STCTRT | NO | Integer | Display groupbufferpool commands |
| DISPLAYPROC | Q9STCTRU | NO | Integer | Display procedure commands |
| STARTPROC | Q9STCTRV | NO | Integer | Start procedure commands |
| STOPPROC | Q9STCTRW | NO | Integer | Stop procedure commands |
| DISPLAYGROUP | Q9STCTRX | NO | Integer | Display group commands |
| ALTERUTIL | Q9STCTRY | NO | Integer | Alter utility commands (DB2 5.1 and later only) |
| DISPLAYFUNC | Q9STCTRZ | NO | Integer | Number of DISPLAY FUNCTION commands (DB2 6.1 and later only) |
| STARTFUNC | Q9STCTX0 | NO | Integer | Number of START FUNCTION commands (DB2 6.1 and later only) |
| STOPFUNC | Q9STCTX1 | NO | Integer | Number of STOP FUNCTION commands (DB2 6.1 and later only) |

Table 3. DMRSTAT Columns (Continued)

| Performance Reporter Column Name | SMF Field Name | Used in Report | Field Type | Description |
|---|---|---|---|---|
| SETLOG | Q9STCTX2 | NO | Integer | Number of SET LOG commands (DB2 6.1 and later only) |
| DISPLAYLOG | Q9STCTX3 | NO | Integer | Number of DISPLAY LOG commands (DB2 6.1 and later only) |
| STATSREASON | QWSDRINV | NO | Integer | Reason to invoke statistics |
| LOGHIGHRBA | QWSDLR | NO | Char(8) | High used log RBA |
| IFIABENDS | QWSDSCA | NO | Integer | IFI abends |
| IFIBADFUNC | QWSDSCU | NO | Integer | IFI unrecognized functions |
| IFICMDREQ | QWSDSCCO | NO | Integer | IFI command requests |
| IFIREADAREQ | QWSDSCRA | NO | Integer | IFI read-A requests |
| IFIREADSREQ | QWSDSCRS | NO | Integer | IFI read-S requests |
| IFIWRITEREQ | QWSDSCWR | NO | Integer | IFI write requests |
| DATACAPTLOGR | QWSDCDLC | NO | Integer | Number of data capture log records retrieved |
| DATACAPTREAD | QWSDCDLR | NO | Integer | Number of data capture log reads |
| DATACAPTDESC | QWSDCDDD | NO | Integer | Number of data capture data descriptions returned |
| DATACAPTDPER | QWSDCDMB | NO | Integer | Number of data capture describes performed |
| DATACAPTABLE | QWSDCDTB | NO | Integer | Number of data capture tables returned to caller |
| DATACAPTLRR | QWSDCDRR | NO | Integer | Data capture log records returned |
| DATACAPTDRR | QWSDCDDR | NO | Integer | Data capture data rows returned |
| PHYSUSP | QVASSUSP | NO | Integer | Physical suspends |
| UNVRES | QVASADUR | NO | Integer | Unavailable resources |
| ADEADLOCK | QVASADDL | NO | Integer | Allocation deadlocks |
| INVRESOURCE | QVASADIR | NO | Integer | Invalid resources |
| SOSCONTRACT | QSSTCONT | NO | Integer | SOS storage contractions |
| SOSDETECT | QSSTCRIT | NO | Integer | SOS detected |
| SOSABEND | QSSTABND | NO | Integer | Abends issued due to SOS |
| DBATINACTCUR | QDSTQCIT | NO | Integer | Current number of TYPE 1 inactive database access threads |

Table 3. DMRSTAT Columns (Continued)

| Performance Reporter Column Name | SMF Field Name | Used in Report | Field Type | Description |
|---|---|---|---|---|
| DBATINACTMAX | QDSTQMIT | NO | Integer | Maximum number of TYPE 1 inactive database access threads that existed |
| DBATACTCUR | QDSTCNAT | NO | Integer | Current number of active database access threads |
| DBATACTMAX | QDSTHWAT | NO | Integer | Maximum number of active database access threads that existed |
| TERMMAXTYPE1 | QDSTNITC | NO | Integer | Number of connections terminated because maximum number of TYPE 1 inactive threads was reached (DB2 6.1 and later only) |
| CURTYPE2INACT | QDSTCIN2 | NO | Integer | Current number of TYPE 2 inactive threads (DB2 6.1 and later only) |
| MAXTYPE2INACT | QDSTMIN2 | NO | Integer | Maximum number of TYPE 2 inactive threads (DB2 6.1 and later only) |
| TYPE2QD | QDSTQIN2 | NO | Integer | Number of queued receive requests for a TYPE 2 inactive thread, plus the number of requests for new connections that were received after the maximum number of remote active threads was reached (DB2 6.1 and later only) |
| CURTYPE2QD | QDSTNQR2 | NO | Integer | Current number of TYPE 2 inactive threads that are queued waiting to process. This includes receive requests that completed for a TYPE 2 inactive thread and requests for new connections that were received after the maximum number of remote active threads was reached (DB2 6.1 and later only) |
| MAXTYPE2QD | QDSTMQR2 | NO | Integer | Maximum number of TYPE 2 inactive threads that were queued waiting for a database access thread. This includes both receive requests that completed for a TYPE 2 inactive thread and requests for new connections that were received after the maximum number of remote active threads was reached (DB2 6.1 and later only) |

Table 3. DMRSTAT Columns (Continued)

| Performance Reporter Column Name | SMF Field Name | Used in Report | Field Type | Description |
|---|---|---|---|---|
| CURDBATFREE | QDSTNADS | NO | Integer | Current number of active database access thread slots that are not in use. A slot is assigned when a queued receive request is completed or a request for a new connection is received (DB2 6.1 and later only) |
| MAXDBATFREE | QDSTMADS | NO | Integer | Maximum number of database access thread slots that were not in use because no processing was required to complete a queued receive request or to establish a new connection (DB2 6.1 and later only) |
| CONCOLDSTART | QDSTCSTR | NO | Integer | Number of cold start connections with all remote locations |
| CONWARMSTART | QDSTWSTR | NO | Integer | Number of warm start connections with all remote locations |
| RESYNATTEMPT | QDSTRSAT | NO | Integer | Number of resynchronization connections attempted with all remote locations |
| RESYNSUCCEED | QDSTRSSU | NO | Integer | Number of resynchronization connections that succeeded with all remote locations |
| CREATEALIAS | QXCRALS | NO | Integer | Number of create alias |
| DROPALIAS | QXDRPAL | NO | Integer | Number of drop alias |
| MULTINDEXYES | QXMIAP | NO | Integer | Number of times RID list processing is used |
| MULTINDEXNOS | QXSMIAP | NO | Integer | Number of times RID list processing could not be used due to no available storage |
| MULTINDEXNOM | QXRMIAP | NO | Integer | Number of times RID list processing could not be used due to exceeded limits |
| SETHOSTV | QXSETHV | NO | Integer | Number of set host variable |
| ALTERDB | QXALDAB | NO | Integer | Number of alter database |
| DROPPKG | QXDRPPKG | NO | Integer | Number of drop package |
| DESCRBTABL | QXDSCRTB | NO | Integer | Number of describe table |
| APALLOCS | QTALLOCA | NO | Integer | Plan allocation attempts |
| SPALLOCS | QTALLOC | NO | Integer | Successful plan allocations |
| BINDINVRES | QTINVRID | NO | Integer | Invalid resource IDs |

Table 3. DMRSTAT Columns (Continued)

| **Performance Reporter Column Name** | **SMF Field Name** | **Used in Report** | **Field Type** | **Description** |
|---|---|---|---|---|
| BINDADD | QTBINDA | NO | Integer | Bind add subcommands |
| BINDREPL | QTBINDR | NO | Integer | Bind replace subcommands |
| TESTBIND | QTTESTB | NO | Integer | Test bind subcommands |
| PLANSBOUND | QTPLNBD | NO | Integer | Number of plans bound |
| REBINDS | QTREBIND | NO | Integer | Rebind subcommands |
| AREBINDS | QTRBINDA | NO | Integer | Rebind attempts |
| SREBINDS | QTPLNRBD | NO | Integer | Successful rebinds |
| BINDFREES | QTFREE | NO | Integer | Free plan subcommands |
| BINDAFREES | QTFREEA | NO | Integer | Free plan attempts |
| BINDSFREES | QTPLNFRD | NO | Integer | Number of plans freed |
| CURROPEN | QTDSOPN | NO | Integer | Number of data sets currently open |
| MAXOPEN | QTMAXDS | NO | Integer | Maximum data sets open concurrently |
| AUTHCKCACHE | QTAUCCH | NO | Integer | Successful authorization checks from cache |
| AUTHCKPUB | QTAUPUB | NO | Integer | Number of successful PUBLIC authority checks |
| MAXPAGBLK | QTMAXPB | NO | Integer | Maximum page set blocks on deferred close queue |
| PKGACAUTH | QTPACAUT | NO | Integer | Number of successful checks for package execute authority made using the package authorization cache including public authorization checks (DB2 5.1 and later only) |
| PKGACPUB | QTPACPUB | NO | Integer | Number of successful checks for package execute authority held by public (DB2 5.1 and later only) |
| PKGACNOT | QTPACNOT | NO | Integer | Number of unsuccessful checks for package execute authority made using the package authorization cache because an applicable entry was not found in the cache (DB2 5.1 and later only) |
| PKGACOWID | QTPACOW1 | NO | Integer | Number of times DB2 overwrote an authorization ID in the cache (DB2 5.1 and later only) |

Table 3. DMRSTAT Columns (Continued)

| Performance Reporter Column Name | SMF Field Name | Used in Report | Field Type | Description |
|---|---|---|---|---|
| PKGACOWENT | QTPACOW2 | NO | Integer | Number of times DB2 overwrote an entry for a package or collection in the cache (DB2 5.1 and later only) |
| REOPEN | QTREOPN | NO | Integer | Reopen from slow close |
| DRAINDSN | QTDSDRN | NO | Integer | Number of data sets closed by slow drain because the deferred close threshold was reached |
| RWROCONVERT | QTPCCT | NO | Integer | Number of data sets converted from R/W to R/O |
| SLOWCLSDD | QTSLWDD | NO | Integer | Number of data sets not currently used, but not closed because of deferred close |
| BINDADDPK | QTBINDPA | NO | Integer | Bind add package |
| BINDREPLPK | QTBINDPR | NO | Integer | Bind replace package |
| PKGBIND | QTPKGBD | NO | Integer | Packages bound |
| REBINDPK | QTRBINDP | NO | Integer | Rebind package |
| REBINDPKA | QTRBNDPA | NO | Integer | Rebind package attempts |
| REBINDPKS | QTPKGRBD | NO | Integer | Rebind package successful |
| FREEPKGC | QTFREEP | NO | Integer | Free package commands |
| FREEPKGA | QTFREEAP | NO | Integer | Free package attempts |
| FREEPKGS | QTPKGFRD | NO | Integer | Free package successful |
| AUTOBINDPKA | QTAUTOBA | NO | Integer | Autobind package attempts |
| AUTOBINDPKS | QTPKABND | NO | Integer | Autobind package successful |
| ALLOCPKGA | QTPKALLA | NO | Integer | Allocate package attempts |
| ALLOCPKGS | QTPKALL | NO | Integer | Allocate package successful |
| UNLOCKREQS | QTXAUNLK | NO | Integer | Unlock requests |
| QUERYREQS | QTXAQRY | NO | Integer | Lock query requests |
| CHNGREQS | QTXACHG | NO | Integer | Lock change requests |

Table 3. DMRSTAT Columns (Continued)

| **Performance Reporter Column Name** | **SMF Field Name** | **Used in Report** | **Field Type** | **Description** |
|---|---|---|---|---|
| IRLMREQS | QTXAIRLM | NO | Integer | Lock other IRLM requests |
| DMRAUTOCUST | none | NO | Integer | DMR column used only as a marker during AutoCustomization; all columns following this one in the create table member are not used in the reports (NO in *Used in Report* column) and can be deleted as a group as an AutoCustomization option |

# DMRSTDF—DDF Statistics Table

DMRSTDF is an optional table for distributed statistics. Each row in the table represents information about one DDF location for one statistics interval.

**Note:** The columns in Table 4 are shown in the same sequence as the SMF record sections. All columns not used (NO in *Used in Report* column below) in the predefined reports are defined at the end of the actual DB2 table after the dummy column, DMRAUTOCUST. These columns can be deleted as a group as an AutoCustomization option, or individually before the table is created. Data sharing columns are defined after the dummy column, DMRACSHARE; they can be deleted as a group as an AutoCustomization option as well.

Table 4. DDF Statistics Columns

| Performance Reporter Column Name | SMF Field Name | Used in Report | Field Type | Description |
|---|---|---|---|---|
| SYSTEMID | SM100SID | YES | Char(4) | System ID (SMF ID) |
| SUBSYSTEM | SM100SSI | YES | Char(4) | Subsystem ID - name of DB2 subsystem |
| DATETIME | QWHSSTCK | YES | Timestamp | Date and time record was created |
| DATE | QWHSSTCK | YES | Date | Date record was created |
| YEAR | QWHSSTCK | YES | Char(4) | Year record was created |
| MONTH | QWHSSTCK | YES | Char(2) | Month record was created |
| DAY | QWHSSTCK | YES | Char(2) | Day record was created |
| TIME | QWHSSTCK | YES | Time | Time record was created |
| HOUR | QWHSSTCK | YES | Char(2) | Hour record was created |
| DAYOFWEEK# | QWHSSTCK | NO | Smallint | Relative day of week, 1 to 7, where Monday=1 and Sunday=7 |
| DAYOFWEEK | QWHSSTCK | NO | Char(3) | MON, TUE, WED, THU, FRI, SAT, SUN |
| WEEK# | QWHSSTCK | NO | Integer | Relative week number |
| LOCATION | QWHSLOCN | YES | Char(16) | Local location name (DB2 subsystem ID if not defined) |
| GROUPNAME | QWHADSGN | YES | Char(8) | Data sharing group name |
| MEMBERNAME | QWHAMEMN | YES | Char(8) | Data sharing member name |
| DDFLOCATION | QLSTLOCN | YES | Char(16) | Name of the remote location |
| SQLSENT | QLSTSQLS | YES | Integer | SQL statements sent |
| SQLRECV | QLSTSQLR | YES | Integer | SQL statements received |
| ROWSENT | QLSTROWS | YES | Integer | Rows sent |
| ROWRECV | QLSTROWR | YES | Integer | Rows received |

Table 4. DDF Statistics Columns

| Performance Reporter Column Name | SMF Field Name | Used in Report | Field Type | Description |
|---|---|---|---|---|
| BYTESENT | QLSTBYTS | NO | Integer | Bytes sent |
| BYTERECV | QLSTBYTR | NO | Integer | Bytes received |
| CONVSENT | QLSTCNVS | YES | Integer | Conversations initiated from this site |
| CONVRECV | QLSTCNVR | YES | Integer | Conversations initiated to this site |
| MSGSSENT | QLSTMSGS | NO | Integer | Messages sent |
| MSGSRECV | QLSTMSGR | NO | Integer | Messages received |
| TRANSENT | QLSTTRNS | YES | Integer | Transactions migrated to remote |
| TRANRECV | QLSTTRNR | YES | Integer | Transactions migrated from remote |
| COMMITSENT | QLSTCOMS | NO | Integer | Commits sent |
| COMMITRECV | QLSTCOMR | NO | Integer | Commits received |
| ABORTSENT | QLSTABRS | NO | Integer | Rollbacks sent |
| ABORTRECV | QLSTABRR | NO | Integer | Rollbacks received |
| CONVQUED | QLSTCNVQ | YES | Integer | Conversation requests queued |
| SWLIMBLK | QLSTCBLB | NO | Integer | Switch to limited block protocol |
| SQLBOUND | QLSTRBND | NO | Integer | SQL bound for remote access |
| ROWSBUFF | QLSTBROW | NO | Integer | Number of rows in buffer |
| BLKSENT | QLSTBTBF | YES | Integer | Blocks sent using block fetch |
| BLKRECV | QLSTBRBF | YES | Integer | Blocks received using block fetch |
| PREPAREQSENT | QLSTPRSE | YES | Integer | Number of PREPARE requests sent to participant |
| PREPAREQSREC | QLSTPRRC | YES | Integer | Number of PREPARE requests received from coordinator |
| LASTAGNTSENT | QLSTLASE | YES | Integer | Number of LAST AGENT requests sent to coordinator |
| LASTAGENTREC | QLSTLARC | YES | Integer | Number of LAST AGENT requests received from initiator |
| COMMITRQSENT | QLSTCRSE | YES | Integer | Number of COMMIT requests sent to participant |
| COMMITREQREC | QLSTCRRC | YES | Integer | Number of COMMIT requests received from coordinator |
| BACKOUTSSENT | QLSTBKSE | YES | Integer | Number of BACKOUT requests sent to participant |

Table 4. DDF Statistics Columns

| Performance Reporter Column Name | SMF Field Name | Used in Report | Field Type | Description |
|---|---|---|---|---|
| BACKOUTRQREC | QLSTBKRC | YES | Integer | Number of BACKOUT requests received from coordinator |
| FORGETRPSENT | QLSTRRSE | NO | Integer | Number of FORGET responses sent to coordinator |
| FORGETRESREC | QLSTRRRC | NO | Integer | Number of FORGET responses received from participant |
| RQCOMMITSENT | QLSTVYSE | YES | Integer | Number of request COMMIT responses sent to coordinator |
| REQCOMMITREC | QLSTVYRC | YES | Integer | Number of request COMMIT responses received from participant |
| BKOUTRSPSENT | QLSTVNSE | NO | Integer | Number of BACKOUT responses sent to coordinator |
| BKOUTRESPREC | QLSTVNRC | NO | Integer | Number of BACKOUT responses received from participant |
| INDOUBTREMCO | QLSTINDT | YES | Integer | Number of threads indoubt with remote location as coordinator |
| COMMITREMTCO | QLSTCPTR | YES | Integer | Number of COMMIT operations with remote location as coordinator |
| ROLLBKREMTCO | QLSTRBTR | YES | Integer | Number of ROLLBACK operations with remote location as coordinator |
| DMRACSHARE | none | NO | Integer | DMR column used only as a marker during AutoCustomization to delete data sharing columns, if requested |
| DMRAUTOCUST | none | NO | Integer | DMR column used only as a marker during AutoCustomization; all columns following this one in the create table member are not used in the reports (NO in *Used in Report* column) and can be deleted as a group as an AutoCustomization option |

## DMRSBFDT—Pool Detail Buffer Statistics Table

DMRSBFDT is an optional table for detailed statistics per buffer pool. One statistics record is created from each pair of SMF 100 records. This record is further processed to create a delta record, showing the changes in values during this statistics interval. Each row in the table represents buffer statistics for one pool for one statistics interval within DB2.

**Note:** The columns in Table 5 are in the same sequence as the SMF record sections. All columns not used (NO in *Used in Report* column below) in the predefined reports are defined at the end of the actual DB2 table after the dummy column, DMRAUTOCUST. These columns can be deleted as a group as an AutoCustomization option, or individually before the table is created. Data sharing columns are defined after the dummy column, DMRACSHARE; they can be deleted as a group as an AutoCustomization option as well.

Table 5. DMRSBFDT Columns

| Performance Reporter Column Name | SMF Field Name | Used in Report | Field Type | Description |
|---|---|---|---|---|
| SYSTEMID | SM100SID | YES | Char(4) | System ID (SMF ID) |
| SUBSYSTEM | SM100SSI | YES | Char(4) | Subsystem ID - name of DB2 subsystem |
| BPOOLID | QBGLGN | NO | Integer | Buffer pool ID |
| DATETIME | QWHSSTCK | YES | Timestamp | Date and time record was created |
| DATE | QWHSSTCK | YES | Date | Date record was created |
| YEAR | QWHSSTCK | YES | Char(4) | Year record was created |
| MONTH | QWHSSTCK | YES | Char(2) | Month record was created |
| DAY | QWHSSTCK | YES | Char(2) | Day record was created |
| TIME | QWHSSTCK | YES | Time | Time record was created |
| HOUR | QWHSSTCK | YES | Char(2) | Hour record was created |
| DAYOFWEEK# | QWHSSTCK | NO | Smallint | Relative day of week, 1 to 7, where Monday=1 and Sunday=7 |
| DAYOFWEEK | QWHSSTCK | NO | Char(3) | MON, TUE, WED, THU, FRI, SAT, SUN |
| WEEK# | QWHSSTCK | NO | Integer | Relative week number |
| LOCATION | QWHSLOCN | YES | Char(16) | Local location name (DB2 subsystem ID if not defined) |
| GROUPNAME | QWHADSGN | YES | Char(8) | Data sharing group name |
| MEMBERNAME | QWHAMEMN | YES | Char(8) | Data sharing member name |
| BPNAME | QBSTPID | NO | Char(6) | BP0 - BP32K9 character format buffer pool ID |
| BPGETPAGE | QBSTGET | YES | Integer | Number of GETPAGES |

Table 5. DMRSBFDT Columns (Continued)

| Performance Reporter Column Name | SMF Field Name | Used in Report | Field Type | Description |
|---|---|---|---|---|
| BPREADS | QBSTRIO | YES | Integer | Synchronous read I/O |
| BPDWV | QBSTDWV | YES | Integer | Number of times the vertical deferred write threshold was reached |
| BPFAILMAX | QBSTXFL | YES | Integer | Expansions failed - pool was full |
| BPFAILSOS | QBSTXFV | YES | Integer | Expansions failed - GETMAIN |
| BPPGUPDAT | QBSTSWS | YES | Integer | Number of page update requests |
| BPPGWRITN | QBSTPWS | YES | Integer | Pages written |
| BPASYNCWR | QBSTWIO | YES | Integer | Asynchronous writes |
| BPACTBUFF | QBSTCBA | YES | Integer | Current active buffers |
| BPPAGEINRD | QBSTRPI | YES | Integer | Pageins for read I/O |
| BPPAGEINWR | QBSTWPI | YES | Integer | Pageins for write I/O |
| BPOPENSOK | QBSTDSO | NO | Integer | Number of successful opens |
| BPSYNCWR | QBSTIMW | YES | Integer | Immediate (synchronous) write I/O |
| BPPFREQS | QBSTSEQ | YES | Integer | Number of sequential prefetch requests |
| BPFPAGESRD | QBSTSPP | YES | Integer | Prefetch pages read |
| BPPFNOBFR | QBSTSPD | YES | Integer | Prefetch disabled - no buffers |
| BPPFNORDN | QBSTREE | YES | Integer | Prefetch disabled - no read engine |
| BPWEUNAVAL | QBSTWEE | YES | Integer | Prefetch disabled - write engine |
| BPDEFWRITE | QBSTDWT | YES | Integer | Times deferred write threshold reached |
| BPDMCRITIC | QBSTDMC | YES | Integer | Times DM critical threshold reached |
| BPMIGDS | QBSTMIG | YES | Integer | Migrated data sets encountered |
| BPRTO | QBSTRTO | YES | Integer | Number of recall timeouts |
| BPPIO | QBSTPIO | YES | Integer | Number of aynchronous read I/Os because of sequential prefetch |
| BPPFNOWKF | QBSTWKPD | YES | Integer | Prefetch abort - zero quantity |
| BPMAX | QBSTMAX | YES | Integer | Number of work files not created because of insufficient buffer resources |
| BPMAXWKFIL | QBSTWFM | YES | Integer | Maximum work files in merge |

Table 5. DMRSBFDT Columns (Continued)

| Performance Reporter Column Name | SMF Field Name | Used in Report | Field Type | Description |
|---|---|---|---|---|
| BPDISTREAD | QBSTWDRP | YES | Integer | Pages for destructive read |
| BPDEQVDWQD | QBSTWBVQ | YES | Integer | Dequeue from VDWQ for destructive read |
| BPMERGPASS | QBSTWFR | YES | Integer | Number of merge passes |
| BPWKFLMERG | QBSTWFT | YES | Integer | Total work files in merge |
| BPWKREJBUF | QBSTWFD | YES | Integer | Work files rejected / no buffer |
| BPMERGNBUF | QBSTWFF | YES | Integer | Merge passes with insufficient buffers |
| BPLISTPREF | QBSTLPF | YES | Integer | List prefetch requests |
| BPDPF | QBSTDPF | YES | Integer | Number of dynamic prefetch requests |
| BPVPA | QBSTVPA | YES | Integer | Number of successful virtual buffer pool expansions or contractions because of the ALTER BUFFERPOOL command |
| BPHPA | QBSTHPA | YES | Integer | Number of successful hiperpool expansions or contractions because of the ALTER BUFFERPOOL command |
| BPHRE | QBSTHRE | YES | Integer | Number of successful synchronous requests to move a page from a hiperpool to a virtual buffer pool |
| BPHRA | QBSTHRA | YES | Integer | Number of pages moved successfully from the hiperpool to the virtual buffer pool asynchronously |
| BPHRF | QBSTHRF | YES | Integer | Number of pages for which a synchronous or asynchronous read request failed because the backing expanded storage page was stolen by the system |
| BPHWR | QBSTHWR | YES | Integer | Number of successful requests issued by DB2 to synchronously move a page from the virtual buffer pool to the hiperpool |
| BPHWA | QBSTHWA | YES | Integer | Number of pages successfully moved asynchronously from the virtual buffer pool to the hiperpool |

Table 5. DMRSBFDT Columns (Continued)

| Performance Reporter Column Name | SMF Field Name | Used in Report | Field Type | Description |
|---|---|---|---|---|
| BPHWF | QBSTHWF | YES | Integer | Number of pages for which a synchronous or asynchronous write request failed because of a shortage of expanded storage |
| BPVPL | QBSTVPL | YES | Integer | Number of buffer pools allocated for a virtual buffer pool |
| BPHPL | QBSTHPL | YES | Integer | Number of buffers allocated for a hiperpool |
| BPDPP | QBSTDPP | YES | Integer | Number of asynchronous page reads because of dynamic prefetch |
| BPLPP | QBSTLPP | YES | Integer | Number of asynchronous page reads because of list prefetch |
| BPDIO | QBSTDIO | YES | Integer | Number of asynchronous read I/Os because of dynamic prefetch |
| BPLIO | QBSTLIO | YES | Integer | Number of asynchronous read I/Os because of list prefetch |
| BPSGT | QBSTSGT | YES | Integer | Number of GETPAGE requests issued by sequential access requesters |
| BPSIO | QBSTSIO | YES | Integer | Number of synchronous read I/Os issued by sequential access requesters |
| BPNGT | QBSTNGT | YES | Integer | Number of times conditional GETPAGE requests could not be satisfied for this buffer pool |
| BPXIS | QBSTXIS | YES | Integer | Highest number of concurrent prefetch I/O streams allocated for supporting queries processed in parallel in this buffer pool |
| BPJIS | QBSTJIS | YES | Integer | Number of requested prefetch I/O streams denied because of a storage shortage in the buffer pool |
| BPPQO | QBSTPQO | YES | Integer | Number of requests made for processing queries in parallel in this buffer pool |
| BPPQF | QBSTPQF | YES | Integer | Number of times during this statistics interval DB2 could not allocate the requested number of buffer pages to allow a parallel group to run to the planned degree |

Table 5. DMRSBFDT Columns (Continued)

| Performance Reporter Column Name | SMF Field Name | Used in Report | Field Type | Description |
|---|---|---|---|---|
| BPPL1 | QBSTPL1 | YES | Integer | Number of occurrences when the prefetch quantity is reduced from normal to one-half of normal |
| BPPL2 | QBSTPL2 | YES | Integer | Number of occurrences when the prefetch quantity is reduced from one-half to one-quarter of normal |
| BPARA | QBSTARA | YES | Integer | Number of pages moved successfully from the hiperpool to the virtual buffer pool by the asynchronous data mover facility |
| BPARF | QBSTARF | YES | Integer | Number of pages for which a read request, using the asynchronous data mover facility, failed because the backing expanded storage was stolen or some other error occurred |
| BPAWA | QBSTAWA | YES | Integer | Number of pages moved successfully from the virtual buffer pool to the hiperpool by the asynchronous data mover facility |
| BPAWF | QBSTAWF | YES | Integer | Number of pages for which a write request, using the asynchronous data mover facility, failed because the backing expanded storage was stolen or some other error occurred |
| BPHBE | QBSTHBE | YES | Integer | Number of hiperpool buffers currently backed by expanded storage |
| BPLPLADDS | QBSTLPL | NO | Integer | Number of times pages added to LPL (DB2 7.1 and later only) |
| DMRACSHARE | none | NO | Integer | DMR column used only as a marker during AutoCustomization to delete data sharing columns, if requested |
| GBPREADINVBD | QBGLXD | NO | Integer | Synchronous coupling facility reads caused by invalid buffer and with data returned |
| GBPREADINVBR | QBGLXR | NO | Integer | Synchronous coupling facility reads caused by invalid buffer with no data returned and a directory entry created |
| GBPREADINVBN | QBGLXN | NO | Integer | Synchronous coupling facility reads caused by invalid buffer with no data returned and no directory entry created |

Table 5. DMRSBFDT Columns (Continued)

| Performance Reporter Column Name | SMF Field Name | Used in Report | Field Type | Description |
|---|---|---|---|---|
| GBPREADNOPGD | QBGLMD | NO | Integer | Synchronous coupling facility reads caused by page not in buffer pool and with data returned |
| GBPREADNOPGR | QBGLMR | NO | Integer | Synchronous coupling facility reads caused by page not in buffer pool with no data returned and a directory entry created |
| GBPREADNOPGN | QBGLMN | NO | Integer | Synchronous coupling facility reads caused by page not in buffer pool with no data returned and no directory entry created |
| GBPWRITCHG | QBGLSW | NO | Integer | Changed pages written synchronously to group buffer pool |
| GBPWRITCLEAN | QBGLWC | NO | Integer | Clean pages written synchronously to group buffer pool |
| GBPCASTCLASS | QBGLCT | NO | Integer | Group buffer pool castout because class threshold detected |
| GBPCASTGBP | QBGLGT | NO | Integer | Group buffer pool castout because group buffer pool threshold detected. |
| GBPAREADD | QBGLAD | NO | Integer | Asynchronous coupling facility reads with data returned |
| GBPAREADR | QBGLAR | NO | Integer | Asynchronous coupling facility reads without data returned and directory |
| GBPAREADN | QBGLAN | NO | Integer | Asynchronous coupling facility reads without data returned and no directory |
| GBPAWRITCHG | QBGLAW | NO | Integer | Changed pages asynchronously written to group buffer pool |
| GBPAWRITCLEAN | QBGLAC | NO | Integer | Clean pages asynchronously written to group buffer pool |
| GBPCASTOPS | QBGLRC | NO | Integer | Castout operations performed |
| GBPNOCASTENG | QBGLCN | NO | Integer | Castout engine not available |
| GBPNOWRTENG | QBGLSU | NO | Integer | Coupling facility write engine not available |
| GBPNOREAD | QBGLRF | NO | Integer | Coupling facility read not complete |
| GBPNOWRITE | QBGLWF | NO | Integer | Coupling facility write not complete |

Table 5. DMRSBFDT Columns (Continued)

| Performance Reporter Column Name | SMF Field Name | Used in Report | Field Type | Description |
|---|---|---|---|---|
| GBPOTHER | QBGLOS | NO | Integer | Other coupling facility requests (DB2 4.1 only) |
| GBPRDSTGST | QBGLOS | NO | Integer | Number of coupling facility requests to read storage statistics (DB2 5.1 and later only) |
| GBPCHKPT | QBGLCK | NO | Integer | Number of group buffer pool checkpoints triggered by this member (DB2 5.1 and later only) |
| GBPREBUILD | QBGLRB | NO | Integer | Number of group buffer pool rebuilds in which this member participated (DB2 5.1 and later only) |
| GBPUNLKCAST | QBGLUN | NO | Integer | Number of coupling facility requests to unlock the castout lock on the pages (DB2 5.1 and later only) |
| GBPRDCASTCL | QBGLCC | NO | Integer | Number of coupling facility requests to read the castout class (DB2 5.1 and later only) |
| GBPRDCASTST | QBGLCS | NO | Integer | Number of coupling facility requests to read the castout statistics (DB2 5.1 and later only) |
| GBPDELETE | QBGLDN | NO | Integer | Number of group buffer pool requests to delete all directory and data entries for a page set or partition (DB2 5.1 and later only) |
| GBPRDDIR | QBGLRD | NO | Integer | Number of coupling facility requests to read directory information (DB2 5.1 and later only) |
| GBPREGPG | QBGLRG | NO | Integer | Number of coupling facility requests to register a page (DB2 5.1 and later only) |
| GBPUNREGPG | QBGLDG | NO | Integer | Number of coupling facility requests to unregister a page (DB2 5.1 and later only) |
| GBPREGPGLST | QBGLAX | NO | Integer | Number of requests to register a page list in the coupling facility |

Table 5. DMRSBFDT Columns (Continued)

| Performance Reporter Column Name | SMF Field Name | Used in Report | Field Type | Description |
|---|---|---|---|---|
| GBPRETVCHP | QBGLAY | NO | Integer | Number of coupling facility reads to retrieve a changed page from the group buffer pool as a result of feedback from the request to register a page list |
| GBPRETVCLP | QBGLAZ | NO | Integer | Number of coupling facility reads to retrieve a clean page from the group buffer pool as a result of feedback from the request to register a page list |
| GBPEXPLICITXI | QBGLEX | NO | Integer | Number of explicit cross-invalidations (DB2 6.1 and later only) |
| GBPWRITCHG2 | QBGL2W | NO | Integer | Number of coupling facility requests to write changed pages to the secondary group buffer pool for duplexing (DB2 6.1 and later only) |
| GBPWRITSTG2 | QBGL2F | NO | Integer | Number of coupling facility requests to write changed pages to the secondary group buffer pool for duplexing that failed due to a lack of storage in the coupling facility (DB2 6.1 and later only) |
| GBPWRITCHK2 | QBGL2S | NO | Integer | Number of completion checks for writes to the secondary GBP that were suspended because the write had not yet completed processing (DB2 6.1 and later only) |
| GBPDELNMLST2 | QBGL2D | NO | Integer | Number of group buffer pool requests to the secondary group buffer pool to delete a list of pages after they have been castout from the primary group buffer pool (DB2 6.1 and later only) |
| GBPRDCASTST2 | QBGL2R | NO | Integer | Number of coupling facility requests to read the castout statistics for the secondary group buffer pool. Issued by group buffer pool structure owner to check for orphaned data entries in the secondary (DB2 6.1 and later only) |

Table 5. DMRSBFDT Columns (Continued)

| Performance Reporter Column Name | SMF Field Name | Used in Report | Field Type | Description |
|---|---|---|---|---|
| GBPDELNM2 | QBGL2N | NO | Integer | Number of group buffer pool requests to delete a page from the secondary group buffer pool. Issued by group buffer pool structure owner to delete orphaned data entries in the secondary as part of the garbage collection logic (DB2 6.1 and later only) |
| GBPASYNPRIM | QBGLHS | NO | Integer | Asynchronous requests for primary GBP (DB2 7.1 and later only) |
| GBPASYNSEC | QBGL2H | NO | Integer | Asynchronous requests for secondary GBP (DB2 6.1 and later only) |
| GBPDEPGETPG | QBGLGG | NO | Integer | Getpages for GBP-dependent pages (DB2 7.1 and later only) |
| GBPPLKSPMAP | QBGLP1 | NO | Integer | Page P-Lock requests for space map pages (DB2 7.1 and later only) |
| GBPPLKDATA | QBGLP2 | NO | Integer | Page P-Lock requests for data pages (DB2 7.1 and later only) |
| GBPPLKIDX | QBGLP3 | NO | Integer | Page P-Lock requests for index leaf pages (DB2 7.1 and later only) |
| GBPPLKUNLK | QBGLU1 | NO | Integer | Page P-Lock unlock requests (DB2 7.1 and later only) |
| GBPPSUSSPMAP | QBGLS1 | NO | Integer | Page P-Lock suspensions for space map pages (DB2 7.1 and later only) |
| GBPPSUSDATA | QBGLS2 | NO | Integer | Page P-Lock suspensions for data pages (DB2 7.1 and later only) |
| GBPPSUSIDX | QBGLS3 | NO | Integer | Page P-Lock suspensions for index leaf pages (DB2 7.1 and later only) |
| GBPPNEGSPMAP | QBGLN1 | NO | Integer | Page P-Lock negotiations for space map pages (DB2 7.1 and later only) |
| GBPPNEGDATA | QBGLN2 | NO | Integer | Page P-Lock negotiations for data pages (DB2 7.1 and later only) |

Table 5. DMRSBFDT Columns (Continued)

| **Performance Reporter Column Name** | **SMF Field Name** | **Used in Report** | **Field Type** | **Description** |
|---|---|---|---|---|
| GBPPNEGIDX | QBGLN3 | NO | Integer | Page P-Lock negotiations for index leaf pages (DB2 7.1 and later only) |
| DMRAUTOCUST | none | NO | Integer | DMR column used only as a marker during AutoCustomization; all columns following this one in the create table member are not used in the reports (NO in *Used in Report* column) and can be deleted as a group as an AutoCustomization option |

# Chapter 17. DB2 Accounting Tables

This chapter provides detailed information about each of the records in the Performance Reporter performance data tables you can use to produce accounting reports.

The detail accounting records are stored in the DMRACDTL table. The summary accounting records are stored in the DMRACSUM table. They are organized as shown in Table 6. The detail and summary buffer pool accounting records are stored in the DMRABDTL and DMRABSUM tables. They are organized as shown in Table 7 on page 511. Refer to Table 8 on page 516 for DDF accounting records.

The package detail accounting records are stored in the DMRAPDTL table. The package summary accounting records are stored in the DMRAPSUM table. They are organized as shown in Table 9 on page 522.

## DMRACxxx—Accounting Detail and Summary Tables

One accounting record is created from each SMF 101 record received. Each row in the DMRACxxx table represents one transaction or thread within DB2.

**Note:** The columns in Table 6 are shown in the same sequence as the SMF record sections. All columns not used (NO in *Used in Report* column below) in the predefined reports are defined at the end of the actual DB2 table after the dummy column, DMRAUTOCUST. These columns can be deleted as a group as an AutoCustomization option, or individually before the table is created. Data sharing columns are defined after the dummy column, DMRACSHARE; they can be deleted as a group as an AutoCustomization option as well.

Table 6. DMRACDTL/DMRACSUM Columns

| Performance Reporter Column Name | SMF Field Name | Used in Report | Field Type | Description |
|---|---|---|---|---|
| SYSTEMID | SM101SID | YES | Char(4) | System ID (SMF ID) |
| SUBSYSTEM | SM101SSI | YES | Char(4) | Subsystem ID (DB2 subsystem name) |
| PLANNAME | QWHCPLAN | YES | Char(8) | Plan name |
| AUTHID | QWHCAID | YES | Char(8) | Authorization ID |
| CONNECTION | QWHCCN | YES | Char(8) | Connection name |
| CORRID | QWHCCV | YES | Char(12) | Correlation ID |
| ORIGPRIMID | QWHCOPID | YES | Char(8) | Original primary authorization ID |
| LUWIDNID | QWHSNID | YES | Char(8) | LUWID - Network ID |
| LUWIDLUNM | QWHSLUNM | YES | Char(8) | LUWID - Logical unit name |
| LUWIDINST | QWHSLUUV | YES | Char(6) | LUWID - Instance ID |
| LUWIDCOMIT | QWHSLUCC | YES | Real | LUWID - Commit count |

Table 6. DMRACDTL/DMRACSUM Columns

| **Performance Reporter Column Name** | **SMF Field Name** | **Used in Report** | **Field Type** | **Description** |
|---|---|---|---|---|
| CONNTYPE | QWHCATYP | YES | Char(8) | Connection type (TSO, DB2CALL, etc.) |
| DATETIME | QWACESC | YES | Timestamp | Date and time record was created |
| DATE | QWACESC | YES | Date | Date record was created |
| YEAR | QWHSSTCK | YES | Char(4) | Year record was created |
| MONTH | QWACESC | YES | Char(2) | Month record was created |
| DAY | QWACESC | YES | Char(2) | Day record was created |
| TIME | QWACESC | YES | Time | Time record was created |
| HOUR | QWACESC | YES | Char(2) | Hour record was created |
| DAYOFWEEK# | QWACESC | NO | Smallint | Relative day of week, 1 to 7, where Monday=1 and Sunday=7 |
| DAYOFWEEK | QWACESC | NO | Char(3) | MON, TUE, WED, THU, FRI, SAT, SUN |
| WEEK# | QWACESC | NO | Integer | Relative week number |
| LOCATION | QWHSLOCN | YES | Char(16) | Local location name (DB2 subsystem ID if not defined) |
| GROUPNAME | QWHADSGN | YES | Char(8) | Data sharing group name |
| MEMBERNAME | QWHAMEMN | YES | Char(8) | Data sharing member name |
| FIRSTPKG | QPACPKID | YES | Char(18) | First package or DBRM executed (planname if accounting class 7 not active) |

Table 6. DMRACDTL/DMRACSUM Columns

| Performance Reporter Column Name | SMF Field Name | Used in Report | Field Type | Description |
|---|---|---|---|---|
| ACCTTOKN | QWHCTOKN | YES | Char(22) | Accounting token for CICS<br>This field applies to the CICS Attachment Facility, RRSAF, and database access threads. For database access threads, this is the value that is received from the requester system. If the connection to the requester system is through DB2 private protocols, this accounting value is identical to the accounting value used at the requester system. If the connection to the requester system is through DRDA protocols, this accounting value is determined from the first 22 bytes of the correlation token (CRRTKN) value of the access relational database (ACCRDB) command received from the requester system during connect processing. |
| ENDUSERID | QWHCEUID | NO | Char(16) | Optional work station end user ID (DB2 5.1 and later)<br>This can be different from the authorization ID used to connect to DB2. This field contains blanks if the client did not supply this information. |
| ENDUSERTX | QWHCEUTX | NO | Char(32) | Optional end user's transaction or application name that identifies the application that is currently running, not the product that is used to run the application (DB2 5.1 and later)<br>This field contains blanks if the client did not supply this information. |
| ENDUSERWN | QWHCEUWN | NO | Char(18) | Optional end user's workstation name (DB2 5.1 and later)<br>This field contains blanks if the client did not supply this information. |
| TRACEMASK | QWHSMTN | NO | Integer | Active trace mask |

Table 6. DMRACDTL/DMRACSUM Columns

| Performance Reporter Column Name | SMF Field Name | Used in Report | Field Type | Description |
|---|---|---|---|---|
| PSTNUMBER | QWHCCV | YES | Char(4) | PST number - IMS only |
| PSBNAME | QWHCCV | YES | Char(8) | PSB name - IMS only |
| CICSTRAN | QWHCCV | YES | Char(4) | Transaction code - CICS only |
| CORRNAME | QWHCCV | YES | Char(8) | User ID, Jobname, CICS TRNID, or IMS PSBNAME |
| REQLOCATION | QWHDRQNM | NO | Char(16) | Distributed transaction requestor location name |
| REQPROD | QWHDPRID | NO | Char(8) | Distributed transaction requestor product ID |
| REQPRODREL | QWHDPRID | NO | Char(8) | VvvRrrMm - version, release, modification level of the requestor for a distributed transaction |
| NETWORKID | QWACNID | YES | Char(16) | Network ID |
| INTERVAL | none | YES | Integer | Interval for the summary accounting table |
| TRANSCNT | none | YES | Integer | Transaction thread count for the summary accounting table |
| IFCIDSEQ# | QWHSISEQ | NO | Integer | IFCID sequence number |
| THDSTART | QWACBSC | YES | Timestamp | Beginning store-clock time |
| ELAPSETOD | QWACBSC,<br>QWACESC | YES | Decimal(15,6) | Elapsed time |
| ELAPSETCB | QWACBJST,<br>QWACEJST | YES | Decimal(15,6) | TCB CPU time |
| ELAPSESRB | QWACBSRB,<br>QWACESRB | YES | Decimal(15,6) | SRB CPU time (not used in DB2 6.1 and later) |
| REASON | QWACRINV | YES | Integer | Reason accounting invoked |
| P2COMMITS | QWACCOMM | YES | Real | Phase 2 or single-phase commits (sync) |
| ABORTS | QWACABRT | YES | Real | Aborts |
| EDB2TOD | QWACASC | YES | Decimal(15,6) | Elapsed DB2 time |
| EDB2TCB | QWACAJST | YES | Decimal(15,6) | TCB CPU in DB2 |
| EDB2SRB | QWACASRB | YES | Decimal(15,6) | SRB CPU in DB2 (not used in DB2 6.1 and later) |

Table 6. DMRACDTL/DMRACSUM Columns

| Performance Reporter Column Name | SMF Field Name | Used in Report | Field Type | Description |
|---|---|---|---|---|
| EWAITIO | QWACAWTI | YES | Decimal(15,6) | Elapsed wait for synchronous I/O<br>**Note:** Log waits are not included in this field for DB2 6.1 and later. |
| EWAITLAL | QWACAWTL | YES | Decimal(15,6) | Elapsed wait for lock or latch |
| ENTEXEVNT | QWACARNA | YES | Real | Number of DB2 exit/entry events |
| WAITEVNT | QWACARNE | YES | Real | Wait events for synchronous I/O<br>**Note:** Log waits are not included in this field for DB2 6.1 and later. |
| WAITREADIO | QWACAWTR | YES | Decimal(15,6) | Elapsed wait for other read I/O |
| WAITWRITEIO | QWACAWTW | YES | Decimal(15,6) | Elapsed wait for other write I/O |
| WAITSYNCEVENT | QWACAWTE | YES | Decimal(15,6) | Elapsed wait for unit switch<br>**Note:** Data set waits are not included in this field for DB2 6.1 and later. |
| WAITARCLOG | QWAXALOG for DB2 6.1 and later<br>QWACALOG for DB2 5.1 and below | YES | Decimal(15,6) | Elapsed wait for archive log quiesce |
| WEVLOCK | QWACARNL | YES | Real | Wait events for lock/latch |
| WEVREAD | QWACARNR | YES | Real | Wait events for other read |
| WEVWRITE | QWACARNW | YES | Real | Wait events for other write |
| WEVSYNCH | QWACARNS | YES | Real | Wait events for unit switch<br>**Note:** Data set waits are not included in this field for DB2 6.1 and later. |
| ARCLOG | QWAXALCT for DB2 6.1 and later<br>QWACALCT for DB2 5.1 and below | YES | Real | Wait events for archive log quiesce |

Table 6. DMRACDTL/DMRACSUM Columns

| Performance Reporter Column Name | SMF Field Name | Used in Report | Field Type | Description |
|---|---|---|---|---|
| DRAINLKRND | QWAXARND for DB2 6.1 and later<br>QWACARND for DB2 5.1 and below | YES | Real | Wait events for drain locks (accounting or monitor class 3) |
| DRAINLKWDR | QWAXAWDR for DB2 6.1 and later<br>QWACAWDR for DB2 5.1 and below | YES | Decimal(15,6) | Elapsed wait time for a drain lock (accounting or monitor class 3) |
| CLAIMLKWCL | QWAXAWCL for DB2 6.1 and later<br>QWACAWCL for DB2 5.1 and below | YES | Decimal(15,6) | Elapsed wait time for a drain when waiting for claims to be released (accounting or monitor class 3) |
| CLAIMLKRNC | QWAXARNC for DB2 6.1 and later<br>QWACARNC for DB2 5.1 and below | YES | Real | Wait events processed for waits for claims to be released (accounting or monitor class 3) |
| ARCHREADWAR | QWAXAWAR for DB2 6.1 and later<br>QWACAWAR for DB2 5.1 and below | YES | Decimal(15,6) | Elapsed wait for an archive read from tape (accounting or monitor class 3) |
| ARCHREADNAR | QWAXANAR for DB2 6.1 and later<br>QWACANAR for DB2 5.1 and below | YES | Real | Wait events processed for archive read (accounting or monitor class 3) |

Table 6. DMRACDTL/DMRACSUM Columns

| Performance Reporter Column Name | SMF Field Name | Used in Report | Field Type | Description |
|---|---|---|---|---|
| OPENCLSELAP | QWAXOCSE | NO | Decimal(15,6) | Accumulated wait time for a synchronous execution unit switch to the DB2 OPEN/CLOSE data set service or the HSM recall service (DB2 6.1 and later only) |
| SYSLGRNGELAP | QWAXSLSE | NO | Decimal(15,6) | Accumulated wait time for a synchronous execution unit switch to the DB2 SYSLGRNG recording service (DB2 6.1 and later only)<br>This service is also sometimes used for level ID checking for down-level detection. |
| DATASETELAP | QWAXDSSE | NO | Decimal(15,6) | Accumulated wait time for a synchronous execution unit switch to the DB2 data space manager services, which include define data set, extend data set, delete data set, reset data set, and VSAM catalog access (DB2 6.1 and later only) |
| OTHERSWELAP | QWAXOTSE | NO | Decimal(15,6) | Accumulated wait time for a synchronous execution unit switch to other DB2 service tasks (DB2 6.1 and later only) |
| OPENCLSEVNT | QWAXOCNS | NO | Real | Number of wait trace events processed for waits for synchronous execution unit switching to the OPEN/CLOSE service (DB2 6.1 and later only) |
| SYSLGRNGEVNT | QWAXSLNS | NO | Real | Number of wait trace events processed for waits for synchronous execution unit switching to the SYSLGRNG recording service (DB2 6.1 and later only) |
| DATASETEVNT | QWAXDSNS | NO | Real | Number of wait trace events processed for waits for synchronous execution unit switching to the data space manager service tasks (DB2 6.1 and later only) |
| OTHERSWEVNT | QWAXOTNS | NO | Real | Number of wait trace events processed for waits for synchronous execution unit switching to other service tasks (DB2 6.1 and later only) |

Table 6. DMRACDTL/DMRACSUM Columns

| **Performance Reporter Column Name** | **SMF Field Name** | **Used in Report** | **Field Type** | **Description** |
|---|---|---|---|---|
| LATCHCNTWTP | QWACAWTP | YES | Decimal(15,6) | Accumulated wait time because of page latch contention (accounting or monitor class 3) |
| LATCHCNTRNH | QWACARNH | YES | Real | Number of wait trace events processed for page latch contention (accounting or monitor class 3) |
| GBLMSGELAP | QWACAWTG | YES | Decimal(15,6) | Data sharing elapsed wait time sending messages |
| GBLMSGEVNT | QWACARNG | YES | Real | Data sharing waits sending messages |
| GBLLOKELAP | QWACAWTJ | YES | Decimal(15,6) | Data sharing elapsed wait time global lock contention |
| GBLLOKEVNT | QWACARNJ | YES | Real | Data sharing waits for global lock contention |
| SPTCB | QWACSPCP | YES | Decimal(15,6) | TCB time processing SQL calls in DB2 SPAS (DB2 4.1 and later) or a WLM address space (DB2 6.1 and later only) |
| SPTCBINDB2 | QWACSPTT | YES | Decimal(15,6) | Stored procedure TCB time in DB2 (accounting class 2) |
| SPEVNT | QWACSPNE | YES | Real | Stored procedure SQL entry or exit events |
| SPWAITELAP | QWACCAST | YES | Decimal(15,6) | Stored procedure elapsed time waiting for TCB |
| SPWAITCNT | QWACCANM | YES | Real | Stored procedure waits for a TCB |
| PARATASKS | QWACPCNT | YES | Real | Parallel tasks or utility subtasks created for an originating parallel task or utility main task |
| PARALLTASKS | QWACPCNT | NO | Integer | 1 if CPU or I/O parallelism used; otherwise 0 |
| CPUSUCONV | QWACSUCV | NO | Integer | CPU service unit conversion factor (DB2 5.1 and later only) |
| LOGWRTEVNT | QWACARLG | NO | Real | Number of wait trace events processed for waits for log write I/O (DB2 6.1 and later only) |
| LOGWRTELAP | QWACAWLG | NO | Decimal(15,6) | Accumulated wait time for log write I/O (DB2 6.1 and later only) |

Table 6. DMRACDTL/DMRACSUM Columns

| Performance Reporter Column Name | SMF Field Name | Used in Report | Field Type | Description |
|---|---|---|---|---|
| WLMSVCCLASS | QWACWLME | NO | Char(8) | MVS workload manager service class name (DB2 6.1 and later only)<br>This field is used only for database access threads on MVS 5.2 or later. In all other cases, this field contains binary zeroes. The WLM service class determines the MVS WLM priority of the work performed by the database access thread. |
| LOGRECORDS | QWACLRN | NO | Real | Number of log records written (DB2 6.1 and later only)<br>The amount of logging for a thread can exceed the amount of logging for units of recovery because the logging for a thread includes logging of actions performed by DB2 on behalf of the thread. |
| LOGBYTES | QWACRAB | NO | Real | Total number of bytes of log records written (DB2 6.1 and later only) |
| FUNCTCB | QWACUDCP | NO | Decimal(15,6) | Accumulated TCB time used to satisfy user-defined function requests processed in a DB2 stored procedures address space or WLM established address space (DB2 6.1 and later only) |
| FUNCSQLTCB | QWACUDTT | NO | Decimal(15,6) | Accumulated TCB time in DB2 for processing SQL statements issued by user-defined functions (DB2 6.1 and later only)<br>This is the time not included in QWACUDCP. |
| FUNCSQLEVNT | QWACUDNE | NO | Real | Number of SQL entry/exit events performed by user-defined functions (DB2 6.1 and later only) |
| FUNCWAIT | QWACUDST | NO | Decimal(15,6) | Total elapsed time spent waiting for an available TCB before the user-defined function could be scheduled (DB2 6.1 and later only) |

Table 6. DMRACDTL/DMRACSUM Columns

| Performance Reporter Column Name | SMF Field Name | Used in Report | Field Type | Description |
|---|---|---|---|---|
| FUNCELAP | QWACUDEA | NO | Decimal(15,6) | Total elapsed time spent in user-defined functions, including time spent executing SQL (DB2 6.1 and later only) |
| FUNCSQLELAP | QWACUDEB | NO | Decimal(15,6) | Total elapsed time spent for user-defined functions to execute SQL (DB2 6.1 and later only) |
| TRIGGERTCB | QWACTRTT | NO | Decimal(15,6) | Accumulated TCB time used while executing under the control of triggers (DB2 6.1 and later only) |
| TRIGGERELAP | QWACTRET | NO | Decimal(15,6) | Accumulated elapsed time used while executing under the control of triggers (DB2 6.1 and later only) |
| PREENCTCB | QWACPECT | NO | Decimal(15,6) | Accumulated TCB time used before the enclave is created (DB2 6.1 and later only) |
| PREENCSQLTCB | QWACPECD | NO | Decimal(15,6) | Accumulated CPU time used for DB2 to process SQL statements before the enclave is created (DB2 6.1 and later only) |
| SPROCELAP | QWACSPEA | NO | Decimal(15,6) | Total elapsed time spent in stored procedures, including time spent executing SQL (DB2 6.1 and later only) |
| SPROCSQLELAP | QWACSPEB | NO | Decimal(15,6) | Total elapsed time spent executing SQL in stored procedures (DB2 6.1 and later only) |
| ENCTRIGGERTCB | QWACTRTE | NO | Decimal(15,6) | Trigger under enclave TCB time (DB2 6.1 and later only) |
| ENCTRIGGERELAP | QWACTREE | NO | Decimal(15,6) | Trigger under enclave elapsed time (DB2 6.1 and later only) |
| ALLIEDCNT | QWHCATYP | NO | Integer | 1 if not DBAT; includes ALLIEDDISTCNT |
| ALLIEDDISTCNT | QWHCATYP | NO | Integer | 1 if not DBAT but QLAC data present |
| DBATCNT | QWHCATYP | NO | Integer | 1 if DBAT and single QLAC; includes DBATDISTCNT |
| DBATDISTCNT | QWHCATYP | NO | Integer | 1 if DBAT and > 1 QLAC |
| NOPKGCNT | QPAC | NO | Integer | No packaging accounting - 1 if no QPACs |

Table 6. DMRACDTL/DMRACSUM Columns

| Performance Reporter Column Name | SMF Field Name | Used in Report | Field Type | Description |
|---|---|---|---|---|
| NORMTERMCNT | QWACRINV | NO | Integer | Normal termination - 1 if QWACRINV = 4-16 |
| ABNORMTERMCNT | QWACRINV | NO | Integer | Abnormal termination - 1 if QWACRINV > 16 |
| SVPOINTREQ | QWACSVPT | NO | Real | Savepoint requests (DB2 6.1 and later only) |
| SVPOINTREL | QWACRLSV | NO | Real | Release savepoint requests (DB2 6.1 and later only) |
| SVPOROLLBK | QWACRBSV | NO | Real | Rollback savepoint requests (DB2 6.1 and later only) |
| WTETAWTK | QWACAWTK | NO | Decimal(15,6) | Wait time for global contention for child L-locks (DB2 7.1 and later only) |
| WTETAWTM | QWACAWTM | NO | Decimal(15,6) | Wait time for global contention for other L-locks (DB2 7.1 and later only) |
| WTETAWTN | QWACAWTN | NO | Decimal(15,6) | Wait time for global contention for pageset/partition P-locks (DB2 7.1 and later only) |
| WTETAWTO | QWACAWTO | NO | Decimal(15,6) | Wait time for global contention for page P-locks (DB2 7.1 and later only) |
| WTETAWTQ | QWACAWTQ | NO | Decimal(15,6) | Wait time for global contention for other P-locks (DB2 7.1 and later only) |
| WTEVARNK | QWACARNK | NO | Real | Number of events with global contention for child L-locks (DB2 7.1 and later only) |
| WTEVARNM | QWACARNM | NO | Real | Number of events with global contention for other L-locks (DB2 7.1 and later only) |
| WTEVARNN | QWACARNN | NO | Real | Number of events with global contention for pageset/partition P-locks (DB2 7.1 and later only) |
| WTEVARNO | QWACARNO | NO | Real | Number of events with global contention for page P-locks (DB2 7.1 and later only) |
| WTEVARNQ | QWACARNQ | NO | Real | Number of events with global contention for other P-locks (DB2 7.1 and later only) |

Table 6. DMRACDTL/DMRACSUM Columns

| Performance Reporter Column Name | SMF Field Name | Used in Report | Field Type | Description |
|---|---|---|---|---|
| WTETAWFC | QWAXAWFC | NO | Decimal(15,6) | Wait time for force-at-commit (DB2 6.1 and later only) |
| WTEVFCCT | QWAXFCCT | NO | Real | Number of events with force-at-commit (DB2 6.1 and later only) |
| WTETIXLT | QWAXIXLT | NO | Decimal(15,6) | Wait time for asynchronous group buffer pool requests (IXLCACHE + IXLFCOMP) (DB2 6.1 and later only) |
| WTEVIXLE | QWAXIXLE | NO | Real | Number of events with asynchronous group buffer pool requests (IXLCACHE + IXLFCOMP) (DB2 6.1 and later only) |
| SETCURPREC | QXSETCPR | NO | Real | SET CURRENT PRECISION statements (DB2 7.1 and later only) |
| DCLGLOBALTT | QXDCLGTT | NO | Real | DECLARE GLOBAL TEMPORARY TABLE statements (DB2 7.1 and later only) |
| PARAGLOBALTT | QXDEGDTT | NO | Real | Parallel groups using DECLARE TEMPORARY TABLE (DB2 6.1 and later only) |
| SELECTS | QXSELECT | YES | Real | Number of SELECT statements |
| INSERTS | QXINSRT | YES | Real | Number of INSERT statements |
| UPDATES | QXUPDTE | YES | Real | Number of UPDATE statements |
| DELETES | QXDELET | YES | Real | Number of DELETE statements |
| DESCRIBES | QXDESC | YES | Real | Number of DESCRIBE statements |
| PREPARES | QXPREP | YES | Real | Number of PREPARE statements |
| OPENS | QXOPEN | YES | Real | Number of OPEN statements |
| FETCHES | QXFETCH | YES | Real | Number of FETCH statements |
| CLOSES | QXCLOSE | YES | Real | Number of CLOSE statements |
| CREATETBL | QXCRTAB | YES | Real | Number of CREATE TABLE statements |
| CREATEINDX | QXCRINX | YES | Real | Number of CREATE INDEX statements |
| CREATETSP | QXCTABS | YES | Real | Number of CREATE TABLESPACE statements |

Table 6. DMRACDTL/DMRACSUM Columns

| Performance Reporter Column Name | SMF Field Name | Used in Report | Field Type | Description |
|---|---|---|---|---|
| CREATESYN | QXCRSYN | YES | Real | Number of CREATE SYNONYM statements |
| CREATEDB | QXCRDAB | YES | Real | Number of CREATE DATABASE statements |
| CREATESG | QXCRSTG | YES | Real | Number of CREATE STORAGE GROUP statements |
| CREATEVU | QXDEFVU | YES | Real | Number of CREATE VIEW statements |
| DROPINDX | QXDRPIX | YES | Real | Number of DROP INDEX statements |
| DROPTBL | QXDRPTA | YES | Real | Number of DROP TABLE statements |
| DROPTSP | QXDRPTS | YES | Real | Number of DROP TABLESPACE statements |
| DROPDB | QXDRPDB | YES | Real | Number of DROP DATABASE statements |
| DROPSYN | QXDRPSY | YES | Real | Number of DROP SYNONYM statements |
| DROPSG | QXDRPST | YES | Real | Number of DROP STORAGE GROUP statements |
| DROPVU | QXDRPVU | YES | Real | Number of DROP VIEW statements |
| ALTERSG | QXALTST | YES | Real | Number of ALTER STORAGE GROUP statements |
| ALTERTSP | QXALTTS | YES | Real | Number of ALTER TABLESPACE statements |
| ALTERTBL | QXALTTA | YES | Real | Number of ALTER TABLE statements |
| ALTERINDX | QXALTIX | YES | Real | Number of ALTER INDEX statements |
| COMMENTON | QXCMTON | YES | Real | Number of COMMENT ON statements |
| LOCKTBL | QXLOCK | YES | Real | Number of LOCK TABLE statements |
| GRANTS | QXGRANT | YES | Real | Number of GRANT statements |
| REVOKES | QXREVOK | YES | Real | Number of REVOKE statements |

Table 6. DMRACDTL/DMRACSUM Columns

| Performance Reporter Column Name | SMF Field Name | Used in Report | Field Type | Description |
|---|---|---|---|---|
| INCRBINDS | QXINCRB | YES | Real | Number of incremental BIND statements |
| LABELON | QXLABON | YES | Real | Number of LABEL ON statements |
| SETSQLID | QXSETSQL | YES | Real | Number of SET SQLID statements |
| CREATEALIAS | QXCRALS | YES | Real | Number of CREATE ALIAS statements |
| DROPALIAS | QXDRPAL | YES | Real | Number of DROP ALIAS statements |
| MULTINDEXYES | QXMIAP | YES | Real | Number of multi-index paths |
| MULTINDEXNOS | QXNSMIAP | YES | Real | Number of multi-index no storage |
| MULTINDEXNOM | QXMRMIAP | YES | Real | Number of multi-index over maximum |
| SETHOSTV | QXSETHV | YES | Real | Number of SET host variable statements |
| ALTERDB | QXALDAB | YES | Real | Number of ALTER DATABASE statements |
| DROPPKG | QXDRPPKG | YES | Real | Number of DROP PACKAGE statements |
| DESCRBTABL | QXDSCRTB | YES | Real | Number of DESCRIBE TABLE statements |
| PARAMAXDEG | QXMAXDEG | YES | Integer | Maximum degree of parallel I/O processing executed among parallel groups |
| PARATOTGRP | QXTOTGRP | YES | Real | Number of parallel groups executed |
| PARADEGCUR | QXDEGCUR | YES | Real | Number of parallel groups that fell back to sequential operation because of a cursor that can be used for UPDATE or DELETE |
| PARADEGESA | QXDEGESA | YES | Real | Number of parallel groups that fell back to sequential operation because of a lack of ESA sort support |
| PARADEGBUF | QXDEGBUF | YES | Real | Number of parallel groups that fell back to sequential operation because of storage shortage or contention on the buffer pool |

Table 6. DMRACDTL/DMRACSUM Columns

| Performance Reporter Column Name | SMF Field Name | Used in Report | Field Type | Description |
|---|---|---|---|---|
| PARAREDGRP | QXREDGRP | YES | Real | Number of parallel groups processed to a parallel degree less than planned because of a storage shortage or contention on the buffer pool |
| PARANORGRP | QXNORGRP | YES | Real | Number of parallel groups executed to the planned parallel degree |
| PARACON1NO | QXCON1 | YES | Real | Number of CONNECT TYPE 1 statements executed |
| PARACON2NO | QXCON2 | YES | Real | Number of CONNECT TYPE 2 statements executed |
| PARARELNO | QXREL | YES | Real | Number of RELEASE statements executed |
| PARASETCON | QXSETCON | YES | Real | Number of SET CONNECTION statements executed |
| PARASETCDG | QXSETCDG | YES | Real | Number of SET CURRENT DEGREE statements executed |
| PARADEGENC | QXDEGENC | YES | Real | Fallback to sequential because MVS/ESA enclave services not available |
| PARARLFDISABLE | QXRLFDPA | YES | Real | Query parallelism disabled by RLF |
| SETCURRULES | QXSETCRL | YES | Real | Set current rules statements executed |
| SQLCALL | QXCALL | YES | Real | SQL call statements executed |
| SQLCALLAB | QXCALLAB | YES | Real | Stored procedure abnormal executions |
| SQLCALLTO | QXCALLTO | YES | Real | Stored procedure time outs |
| SQLCALLRJ | QXCALLRJ | YES | Real | SQL call statements rejected |

Table 6. DMRACDTL/DMRACSUM Columns

| **Performance Reporter Column Name** | **SMF Field Name** | **Used in Report** | **Field Type** | **Description** |
|---|---|---|---|---|
| PARACOORNO | QXCOORNO | NO | Real | Total number of parallel groups executed on a single DB2 due to one of the following reasons:<br>• When the plan or package was bound, the coordinator subsystem parameter was set to YES, but the parameter is set to NO when the program runs<br>• The plan or package was bound on a DB2 with the coordinator subsystem parameter set to YES, but the program is being run on a different DB2 that has the coordinator value set to NO<br>(DB2 5.1 and later only) |
| PARAISORR | QXISORR | NO | Real | Total number of parallel groups executed on a single DB2 because the plan or package was bound with an isolation value of repeatable read (DB2 5.1 and later only) |
| SQLCRGTT | QXCRGTT | NO | Real | Number of CREATE GLOBAL TEMPORARY TABLE statements (DB2 5.1 and later only) |
| REOPTIMIZE | QXSTREOP | NO | Real | Number of times reoptimization occurred (DB2 5.1 and later only) |
| PARAXDSGRP | QXXCBPNX | NO | Real | Number of parallel groups DB2 intended to run across the data sharing group (DB2 5.1 and later only) |
| PARACSKIP | QXXCSKIP | NO | Real | Number of times the parallelism coordinator had to bypass a DB2 when distributing tasks because there was not enough buffer pool storage on one or more DB2 members (DB2 5.1 and later only) |
| ASSOCLOCR | QXALOCL | NO | Real | Number of associate locator statements executed (DB2 5.1 and later only) |
| ALLOCCUR | QXALOCC | NO | Real | Number of allocate cursor statements executed (DB2 5.1 and later only) |

Table 6. DMRACDTL/DMRACSUM Columns

| **Performance Reporter Column Name** | **SMF Field Name** | **Used in Report** | **Field Type** | **Description** |
|---|---|---|---|---|
| PREPFND | QXSTFND | NO | Real | Number of times DB2 satisfied a PREPARE request by making a copy of a statement in the prepared statement cache (DB2 5.1 and later only) |
| PREPNOTFND | QXSTNFND | NO | Real | Number of times DB2 searched the prepared statement cache but could not find a suitable prepared statement (DB2 5.1 and later only) |
| PREPIMPLICIT | QXSTIPRP | NO | Real | Number of times DB2 did an implicit PREPARE for a statement bound with KEEPDYNAMIC(YES) because the prepared statement cache did not contain a valid copy of the prepared statement (DB2 5.1 and later only) |
| PREPNOIMPLICIT | QXSTNPRP | NO | Real | Number of times DB2 did not PREPARE a statement bound with KEEPDYNAMIC(YES) because the prepared statement cache contained a valid copy of the prepared statement (DB2 5.1 and later only) |
| PREPDISCMAX | QXSTDEXP | NO | Real | Number of times DB2 discarded a prepared statement from the prepared statement cache because the number of prepared statements in the cache exceeded the value of subsystem parameter MAXKEEPD (DB2 5.1 and later only) |
| PREPDISCPROG | QXSTDINV | NO | Real | Number of times DB2 discarded a prepared statement from the prepared statement cache because a program executed a DROP, ALTER, or REVOKE statement against a dependent object (DB2 5.1 and later only) |
| RENAMETBL | QXRNTAB | NO | Real | Number of RENAME TABLE statements (DB2 5.1 and later only) |
| CREATETRIGGER | QXCTRIG | NO | Real | Number of SQL CREATE TRIGGER statements (DB2 6.1 and later only) |
| DROPTRIGGER | QXDRPTR | NO | Real | Number of SQL DROP TRIGGER statements (DB2 6.1 and later only) |

Table 6. DMRACDTL/DMRACSUM Columns

| Performance Reporter Column Name | SMF Field Name | Used in Report | Field Type | Description |
|---|---|---|---|---|
| SETCURRPATH | QXSETPTH | NO | Real | Number of SQL SET CURRENT PATH statements (DB2 6.1 and later only) |
| DROPUDF | QXDRPFN | NO | Real | Number of DROP UDF statements (DB2 6.1 and later only) |
| DROPPROC | QXDRPPR | NO | Real | Number of DROP PROCEDURE statements (DB2 6.1 and later only) |
| CREATEDISTINCT | QXCDIST | NO | Real | Number of CREATE DISTINCT TYPE statements (DB2 6.1 and later only) |
| DROPDISTINCT | QXDDIST | NO | Real | Number of DROP DISTINCT TYPE statements (DB2 6.1 and later only) |
| CREATEFUNC | QXCRUDF | NO | Real | Number of CREATE FUNCTION statements (DB2 6.1 and later only) |
| CREATEPROC | QXCRPRO | NO | Real | Number of CREATE PROCEDURE statements (DB2 6.1 and later only) |
| HOLDLOCATOR | QXHOLDL | NO | Real | Number of HOLD LOCATOR statements (DB2 6.1 and later only) |
| FREELOCATOR | QXFREEL | NO | Real | Number of FREE LOCATOR statements (DB2 6.1 and later only) |
| PARACONFIG | QXREPOP1 | YES | Real | Number of parallel groups for which DB2 reformulated the parallel portion of the access path because the SYSPLEX configuration at run time was different from the SYSPLEX configuration at bind time. This counter is incremented only by the parallelism coordinator at run time (DB2 6.1 and later only) |
| PARANOBP | QXREPOP2 | YES | Real | Number of parallel groups for which DB2 reformulated the parallel portion of the access path because there was not enough buffer pool resource. This counter is incremented only by the parallelism coordinator at run time (DB2 6.1 and later only) |
| CREATEAUXTBL | QXCRATB | NO | Real | Number of CREATE AUXILIARY TABLE statements (DB2 6.1 and later only) |

Table 6. DMRACDTL/DMRACSUM Columns

| Performance Reporter Column Name | SMF Field Name | Used in Report | Field Type | Description |
|---|---|---|---|---|
| MAXLOBSTG | QXSTLOBV | NO | Real | Maximum storage used for LOB values, in megabytes (DB2 6.1 and later only) |
| ALTERFUNC | QXALUDF | NO | Real | Number of ALTER FUNCTION statements (DB2 6.1 and later only) |
| ALTERPROC | QXALPRO | NO | Real | Number of ALTER PROCEDURE statements (DB2 6.1 and later only) |
| DIRECTROW | QXROIMAT | NO | Real | Number of times that DB2 used direct row access to locate a record (DB2 6.1 and later only) |
| DIRECTROWIX | QXROIIDX | NO | Real | Number of times that DB2 attempted to use direct row access but reverted to using an index to locate a record (DB2 6.1 and later only) |
| DIRECTROWTS | QXROITS | NO | Real | Number of times that DB2 attempted to use direct row access but reverted to using a table space scan to locate a record (DB2 6.1 and later only) |
| STMTTRIGGER | QXSTTRG | NO | Real | Number of times a statement trigger is activated (DB2 6.1 and later only) |
| ROWTRIGGER | QXROWTRG | NO | Real | Number of times a row trigger is activated (DB2 6.1 and later only) |
| SQLERRTRIGGER | QXTRGERR | NO | Real | Number of times an SQL error occurred during the execution of a triggered action (DB2 6.1 and later only) |
| MAXSQLCASCADE | QXCASCDP | NO | Real | Maximum level of nested SQL cascading due to triggers, user-defined functions, and stored procedures (DB2 6.1 and later only) |
| FUNC | QXCAUD | NO | Real | Number of user-defined functions executed (DB2 6.1 and later only) |
| FUNCAB | QXCAUDAB | NO | Real | Number of times a user-defined function abended (DB2 6.1 and later only) |
| FUNCTO | QXCAUDTO | NO | Real | Number of times a user-defined function timed out waiting to be scheduled (DB2 6.1 and later only) |

Table 6. DMRACDTL/DMRACSUM Columns

| Performance Reporter Column Name | SMF Field Name | Used in Report | Field Type | Description |
|---|---|---|---|---|
| FUNCRJ | QXCAUDRJ | NO | Real | Number of times a user-defined function was rejected (DB2 6.1 and later only) |
| BPGETPAGE | QBACGET | YES | Real | GETPAGEs |
| BPPGUPDAT | QBACSWS | YES | Real | Pages updated |
| BPSYNCRD | QBACRIO | YES | Real | Synchronous read I/O |
| BPPREFET | QBACSEQ | YES | Real | Sequential prefetch |
| BPSYNCWR | QBACIMW | YES | Real | Synchronous write I/O |
| BPLISTPREF | QBACLPF | YES | Real | Number of list prefetch requests |
| BPDPF | QBACDPF | YES | Real | Number of dynamic prefetch requests |
| BPHRE | QBACHRE | YES | Real | Number of successful hiperpool reads |
| BPHRF | QBACHRF | YES | Real | Number of unsuccessful hiperpool reads |
| BPHWR | QBACHWR | YES | Real | Number of successful hiperpool writes |
| BPHWF | QBACHWF | YES | Real | Number of unsuccessful hiperpool writes |
| BPNGT | QBACNGT | YES | Real | Number of unsuccessful GETPAGE operations |
| BPSIO | QBACSIO | YES | Real | Number of asynchronous pages read by prefetch under the control of the agent |
| BPHPG | QBACHPG | YES | Real | Number of pages moved from a hiperpool to a virtual buffer pool because of a prefetch under the control of the agent |
| DEADLOCKS | QTXADEA | YES | Real | Deadlocks |
| SUSPENDS | QTXASLOC | YES | Real | Suspends |
| TIMEOUTS | QTXATIM | YES | Real | Timeouts |
| LOCKESHR | QTXALES | YES | Real | Lock escalations to shared |
| LOCKEXCL | QTXALEX | YES | Real | Lock escalations to exclusive |
| MAXPGLOCKS | QTXANPL | YES | Integer | Maximum page locks |
| SUSPLATCH | QTXASLAT | YES | Real | Latch suspends |

Table 6. DMRACDTL/DMRACSUM Columns

| Performance Reporter Column Name | SMF Field Name | Used in Report | Field Type | Description |
|---|---|---|---|---|
| SUSPOTHER | QTXASOTH | YES | Real | Other suspends |
| LOCKREQS | QTXALOCK | YES | Real | Lock requests |
| CLAIMREQ | QTXACLNO | YES | Real | Number of claim requests |
| CLAIMREQUN | QTXACLUN | YES | Real | Number of unsuccessful claim requests |
| DRAINREQ | QTXADRNO | YES | Real | Number of drain requests |
| DRAINREQUN | QTXADRUN | YES | Real | Number of unsuccessful drain requests |
| DB2MBRMAX | QWDAXCLM | NO | Integer | Largest number of DB2 members that participated in processing a query (DB2 5.1 and later only) |
| MBRCORR1-8 | QWDAXCQO | NO | Char(8) | For a parallel task, this is a correlating value that indicates the member name of the parallelism coordinator (DB2 5.1 and later only) |
| DMRACSHARE | none | NO | Integer | DMR column used only as a marker during AutoCustomization to delete data sharing columns, if requested |
| GBPREADINVBD | QBGAXD | NO | Real | Synchronous coupling facility reads caused by invalid buffer and with data returned |
| GBPREADINVBR | QBGAXR | NO | Real | Synchronous coupling facility reads caused by invalid buffer with no data returned and a directory entry created |
| GBPREADINVBN | QBGAXN | NO | Real | Synchronous coupling facility reads caused by invalid buffer with no data returned and no directory entry created |
| GBPREADNOPGD | QBGAMD | NO | Real | Synchronous coupling facility reads caused by page not in buffer pool and with data returned |
| GBPREADNOPGR | QBGAMR | NO | Real | Synchronous coupling facility reads caused by page not in buffer pool with no data returned and a directory entry created |
| GBPREADNOPGN | QBGAMN | NO | Real | Synchronous coupling facility reads caused by page not in buffer pool with no data returned and no directory entry created |

Table 6. DMRACDTL/DMRACSUM Columns

| Performance Reporter Column Name | SMF Field Name | Used in Report | Field Type | Description |
|---|---|---|---|---|
| GBPWRITCHG | QBGASW | NO | Real | Changed pages written synchronously to group buffer pool |
| GBPWRITCLEAN | QBGAWC | NO | Real | Clean pages written synchronously to group buffer pool |
| GBPUNREGPG | QBGADG | NO | Real | Number of coupling facility requests to unregister a page (DB2 5.1 and later only) |
| GBPEXPLICITXI | QBGAEX | YES | Real | Number of explicit cross-invalidations (DB2 6.1 and later only) |
| GBPWRITCHG2 | QBGA2W | YES | Real | Number of coupling facility requests to write changed pages to the secondary group buffer pool for duplexing (DB2 6.1 and later only) |
| GBPWRITCHK2 | QBGA2S | NO | Real | Number of completion checks for writes to the secondary GBP that were suspended because the write had not yet completed processing (DB2 6.1 and later only) |
| GBPASYNPRIM | QBGAHS | NO | Real | Asynchronous requests for primary GBP (DB2 7.1 and later only) |
| GBPASYNSEC | QBGA2H | NO | Real | Asynchronous requests for secondary GBP (DB2 6.1 and later only) |
| GBPDEPGETPG | QBGAGG | NO | Real | Getpages for GBP-dependent pages (DB2 7.1 and later only) |
| GBPPLKSPMAP | QBGAP1 | NO | Real | Page P-Lock requests for space map pages (DB2 7.1 and later only) |
| GBPPLKDATA | QBGAP2 | NO | Real | Page P-Lock requests for data pages (DB2 7.1 and later only) |
| GBPPLKIDX | QBGAP3 | NO | Real | Page P-Lock requests for index leaf pages (DB2 7.1 and later only) |
| GBPPLKUNLK | QBGAU1 | NO | Real | Page P-Lock unlock requests (DB2 7.1 and later only) |
| GBPPSUSSPMAP | QBGAS1 | NO | Real | Page P-Lock suspensions for space map pages (DB2 7.1 and later only) |
| GBPPSUSDATA | QBGAS2 | NO | Real | Page P-Lock suspensions for data pages (DB2 7.1 and later only) |
| GBPPSUSIDX | QBGAS3 | NO | Real | Page P-Lock suspensions for index leaf pages (DB2 7.1 and later only) |

Table 6. DMRACDTL/DMRACSUM Columns

| Performance Reporter Column Name | SMF Field Name | Used in Report | Field Type | Description |
|---|---|---|---|---|
| GLPLOCKLK | QTGALPLK | NO | Real | Lock requests for P-Locks |
| GLPLOCKCHG | QTGACPLK | NO | Real | Change requests for P-Locks |
| GLPLOCKUNLK | QTGAUPLK | NO | Real | Unlock requests for P-Locks |
| GLXESSYNCUNLK | QTGALSLM | NO | Real | Lock requests propagated to MVS XES synchronously |
| GLXESSYNCCHG | QTGACSLM | NO | Real | Change requests propagated to MVS XES synchronously |
| GLXESSYNCUNLK | QTGAUSLM | NO | Real | Unlock requests propagated to MVS XES synchronously |
| GLSUSPIRLM | QTGAIGLO | NO | Real | Suspends caused by IRLM global resource contention |
| GLSUSPXES | QTGASGLO | NO | Real | Suspends caused by MVS XES global resource contention |
| GLSUSPFALSE | QTGAFLSE | NO | Real | Suspends caused by false contentions |
| GLINCOMPAT | QTGADRTA | NO | Real | Global lock or change requests denied for incompatible retained lock |
| GLNOTFYSENT | QTGANTFY | NO | Real | Notify messages sent |
| DMRAUTOCUST | none | NO | Integer | DMR column used only as a marker during AutoCustomization; all columns following this one in the create table member are not used in the reports (NO in *Used in Report* column) and can be deleted as a group as an AutoCustomization option |
| RLFTABLEID | QTXARLID | NO | Char(2) | Resource limit table ID |
| RLFLIMDET | QTXAPREC | NO | Integer | Resource limit determination |
| RLFSULIMIT | QTXASLMT | NO | Integer | Limit in SUs |
| RLFCPULIMIT | QTXACLMT | NO | Integer | RLF limit in CPU milliseconds |
| RLFCPULIMITU | QTXACLMT | NO | Decimal(15,6) | RLF limit in CPU seconds |
| RLFCPUUSED | QTXACHUS | NO | Integer | Highest CPU milliseconds used |
| RLFCPUUSEDU | QTXACHUS | NO | Decimal(15,6) | Highest CPU seconds used |
| UNLOCKREQS | QTXAUNLK | NO | Real | Unlock requests |
| QUERYREQS | QTXAQRY | NO | Real | Lock query requests |

Table 6. DMRACDTL/DMRACSUM Columns

| Performance Reporter Column Name | SMF Field Name | Used in Report | Field Type | Description |
|---|---|---|---|---|
| CHNGREQS | QTXACHG | NO | Real | Lock change requests |
| IRLMREQS | QTXAIRLM | NO | Real | Other IRLM requests |
| IFIELAPSED | QIFAAIET | NO | Decimal(15,6) | Accumulated elapsed time spent processing IFI calls |
| IFITCBCPU | QIFAAITT | NO | Decimal(15,6) | Accumulated TCB CPU time spent processing IFI calls |
| IFIELAPDTC | QIFAAMBT | NO | Decimal(15,6) | Accumulated elapsed time spent processing data capture describes |
| IFIELAPEXT | QIFAAMLT | NO | Decimal(15,6) | Accumulated elapsed time spent extracting log records for tables defined with data capture changes |
| IFIENTRIES | QIFAANIF | NO | Real | Number of entries to and exits from IFI events |
| IFILOGREAD | QIFAANLR | NO | Real | Number of log reads performed for processing IFI read requests for IFCID 0185 |
| IFILOGREC | QIFAANRC | NO | Real | Number of log records written for tables defined with data capture changes |
| IFILOGRTN | QIFAANRR | NO | Real | Number of log records returned to caller of IFI reads call for IFCID 0185 |
| IFIDATAROW | QIFAANDR | NO | Real | Number of data rows returned in IFCID 0185 |
| IFIDATADSC | QIFAANDD | NO | Real | Number of data descriptions returned in IFCID 0185 |
| IFIDATACAP | QIFAANMB | NO | Real | Number of data capture describes for processing read requests for IFCID 0185 data |
| IFITABLRTN | QIFAANTB | NO | Real | Number of tables returned to caller of IFI reads call for IFCID 0185 |

## DMRABxxx—Accounting Buffer Detail and Summary Tables

Buffer accounting detail records are created from SMF 101 records that were created with accounting class 7/8 active.

One accounting record is created from each SMF 101 record received. Each row in the DMRABxxx table represents one transaction or thread within DB2.

**Note:** The columns in Table 7 are shown in the same sequence as the SMF record sections. All columns not used (NO in *Used in Report* column below) in the predefined reports are defined at the end of the actual DB2 table after the dummy column, DMRAUTOCUST. These columns can be deleted as a group as an AutoCustomization option, or individually before the table is created. Data sharing columns are defined after the dummy column, DMRACSHARE; they can be deleted as a group as an AutoCustomization option as well.

Table 7. DMRABDTL/DMRABSUM Columns

| Performance Reporter Column Name | SMF Field Name | Used in Report | Field Type | Description |
|---|---|---|---|---|
| SYSTEMID | SM101SID | YES | Char(4) | System ID (SMF ID) |
| SUBSYSTEM | SM101SSI | YES | Char(4) | Subsystem ID (DB2 subsystem name) |
| BPOOLID | QBACPID | YES | Integer | Buffer Pool ID |
| PLANNAME | QWHCPLAN | YES | Char(8) | Plan name |
| AUTHID | QWHCAID | YES | Char(8) | Authorization ID |
| CONNECTION | QWHCCN | YES | Char(8) | Connection name |
| CORRID | QWHCCV | YES | Char(12) | Correlation ID |
| ORIGPRIMID | QWHCOPID | YES | Char(8) | Original primary authorization ID |
| LUWIDNID | QWHSNID | YES | Char(8) | LUWID - Network ID |
| LUWIDLUNM | QWHSLUNM | YES | Char(8) | LUWID - Logical unit name |
| LUWIDINST | QWHSLUUV | YES | Char(6) | LUWID - Instance ID |
| LUWIDCOMIT | QWHSLUCC | YES | Real | LUWID - Commit count |
| CONNTYPE | QWHCATYP | YES | Char(8) | Connection type (TSO, DB2CALL, etc.) |
| DATETIME | QWACESC | YES | Timestamp | Date and time record was created |
| DATE | QWACESC | YES | Date | Date record was created |
| YEAR | QWHSSTCK | YES | Char(4) | Year record was created |
| MONTH | QWACESC | YES | Char(2) | Month record was created |
| DAY | QWACESC | YES | Char(2) | Day record was created |
| TIME | QWACESC | YES | Time | Time record was created |

Table 7. DMRABDTL/DMRABSUM Columns (Continued)

| Performance Reporter Column Name | SMF Field Name | Used in Report | Field Type | Description |
|---|---|---|---|---|
| HOUR | QWACESC | YES | Char(2) | Hour record was created |
| DAYOFWEEK# | QWACESC | NO | Smallint | Relative day of week, 1 to 7, where Monday=1 and Sunday=7 |
| DAYOFWEEK | QWACESC | NO | Char(3) | MON, TUE, WED, THU, FRI, SAT, SUN |
| WEEK# | QWACESC | NO | Integer | Relative week number |
| CPUSUCONV | QWACSUCV | NO | Integer | CPU service unit conversion factor (DB2 5.1 and later only) |
| LOCATION | QWHSLOCN | YES | Char(16) | Local location name (DB2 subsystem ID if not defined) |
| GROUPNAME | QWHADSGN | YES | Char(8) | Data sharing group name |
| MEMBERNAME | QWHAMEMN | YES | Char(8) | Data sharing member name |
| FIRSTPKG | QPACPKID | YES | Char(18) | First package or DBRM executed (planname if accounting class 7 not active) |
| ENDUSERID | QWHCEUID | NO | Char(16) | Optional work station end user ID (DB2 5.1 and later)<br>This can be different from the authorization ID used to connect to DB2. This field contains blanks if the client did not supply this information. |
| ENDUSERTX | QWHCEUTX | NO | Char(32) | Optional end user's transaction or application name that identifies the application that is currently running, not the product that is used to run the application (DB2 5.1 and later)<br>This field contains blanks if the client did not supply this information. |
| ENDUSERWN | QWHCEUWN | NO | Char(18) | Optional end user's workstation name (DB2 5.1 and later)<br>This field contains blanks if the client did not supply this information. |
| TRACEMASK | QWHSMTN | NO | Integer | Active trace mask |
| PSTNUMBER | QWHCCV | YES | Char(4) | PST number - IMS only |

Table 7. DMRABDTL/DMRABSUM Columns (Continued)

| **Performance Reporter Column Name** | **SMF Field Name** | **Used in Report** | **Field Type** | **Description** |
|---|---|---|---|---|
| PSBNAME | QWHCCV | YES | Char(8) | PSB name - IMS only |
| CICSTRAN | QWHCCV | YES | Char(4) | Transaction code - CICS only |
| CORRNAME | QWHCCV | YES | Char(8) | User ID, Jobname, CICS TRNID, or IMS PSBNAME |
| REQLOCATION | QWHDRQNM | NO | Char(16) | Distributed transaction requestor location name |
| REQPROD | QWHDPRID | NO | Char(8) | Distributed transaction requestor product ID |
| REQPRODREL | QWHDPRID | NO | Char(8) | VvvRrrMm - version, release, modification level of the requestor for a distributed transaction |
| NETWORKID | QWACNID | YES | Char(16) | Network ID |
| INTERVAL | none | YES | Integer | Interval for the summary accounting table |
| TRANSCNT | none | YES | Integer | Transaction thread count for the summary accounting table |
| IFCIDSEQ# | QWHSISEQ | NO | Integer | IFCID sequence number |
| BPNAME | QBACPID | YES | Char(6) | BP0 through BP32K9 character format buffer pool ID |
| BPGETPAGE | QBACGET | YES | Real | GETPAGEs |
| BPPGUPDAT | QBACSWS | YES | Real | Pages updated |
| BPSYNCRD | QBACRIO | YES | Real | Synchronous read I/O |
| BPPREFET | QBACSEQ | YES | Real | Sequential prefetch |
| BPSYNCWR | QBACIMW | YES | Real | Synchronous write I/O |
| BPLISTPREF | QBACLPF | YES | Real | Number of list prefetch requests |
| BPDPF | QBACDPF | YES | Real | Number of dynamic prefetch requests |
| BPHRE | QBACHRE | YES | Real | Number of successful hiperpool reads |
| BPHRF | QBACHRF | YES | Real | Number of unsuccessful hiperpool reads |
| BPHWR | QBACHWR | YES | Real | Number of successful hiperpool writes |
| BPHWF | QBACHWF | YES | Real | Number of unsuccessful hiperpool writes |

Table 7. DMRABDTL/DMRABSUM Columns (Continued)

| **Performance Reporter Column Name** | **SMF Field Name** | **Used in Report** | **Field Type** | **Description** |
|---|---|---|---|---|
| BPNGT | QBACNGT | YES | Real | Number of unsuccessful GETPAGE operations |
| BPSIO | QBACSIO | YES | Real | Number of asynchronous pages read by prefetch under the control of the agent |
| BPHPG | QBACHPG | YES | Real | Number of pages moved from a hiperpool to a virtual buffer pool because of a prefetch under the control of the agent |
| DMRACSHARE | none | NO | Integer | DMR column used only as a marker during AutoCustomization to delete data sharing columns, if requested |
| GBPREADINVBD | QBGAXD | NO | Real | Synchronous coupling facility reads caused by invalid buffer and with data returned |
| GBPREADINVBR | QBGAXR | NO | Real | Synchronous coupling facility reads caused by invalid buffer with no data returned and a directory entry created |
| GBPREADINVBN | QBGAXN | NO | Real | Synchronous coupling facility reads caused by invalid buffer with no data returned and no directory entry created |
| GBPREADNOPGD | QBGAMD | NO | Real | Synchronous coupling facility reads caused by page not in buffer pool and with data returned |
| GBPREADNOPGR | QBGAMR | NO | Real | Synchronous coupling facility reads caused by page not in buffer pool with no data returned and a directory entry created |
| GBPREADNOPGN | QBGAMN | NO | Real | Synchronous coupling facility reads caused by page not in buffer pool with no data returned and no directory entry created |
| GBPWRITCHG | QBGASW | NO | Real | Changed pages written synchronously to group buffer pool |
| GBPWRITCLEAN | QBGAWC | NO | Real | Clean pages written synchronously to group buffer pool |
| GBPUNREGPG | QBGADG | NO | Real | Number of coupling facility requests to unregister a page (DB2 5.1 and later only) |

Table 7. DMRABDTL/DMRABSUM Columns (Continued)

| Performance Reporter Column Name | SMF Field Name | Used in Report | Field Type | Description |
|---|---|---|---|---|
| GBPEXPLICITXI | QBGAEX | YES | Real | Number of explicit cross-invalidations (DB2 6.1 and later only) |
| GBPWRITCHG2 | QBGA2W | YES | Real | Number of coupling facility requests to write changed pages to the secondary group buffer pool for duplexing (DB2 6.1 and later only) |
| GBPWRITCHK2 | QBGA2S | NO | Real | Number of completion checks for writes to the secondary GBP that were suspended because the write had not yet completed processing (DB2 6.1 and later only) |
| GBPASYNPRIM | QBGAHS | NO | Real | Asynchronous requests for primary GBP (DB2 7.1 and later only) |
| GBPASYNSEC | QBGA2H | NO | Real | Asynchronous requests for secondary GBP (DB2 6.1 and later only) |
| GBPDEPGETPG | QBGAGG | NO | Real | Getpages for GBP-dependent pages (DB2 7.1 and later only) |
| GBPPLKSPMAP | QBGAP1 | NO | Real | Page P-Lock requests for space map pages (DB2 7.1 and later only) |
| GBPPLKDATA | QBGAP2 | NO | Real | Page P-Lock requests for data pages (DB2 7.1 and later only) |
| GBPPLKIDX | QBGAP3 | NO | Real | Page P-Lock requests for index leaf pages (DB2 7.1 and later only) |
| GBPPLKUNLK | QBGAU1 | NO | Real | Page P-Lock unlock requests (DB2 7.1 and later only) |
| GBPPSUSSPMAP | QBGAS1 | NO | Real | Page P-Lock suspensions for space map pages (DB2 7.1 and later only) |
| GBPPSUSDATA | QBGAS2 | NO | Real | Page P-Lock suspensions for data pages (DB2 7.1 and later only) |
| GBPPSUSIDX | QBGAS3 | NO | Real | Page P-Lock suspensions for index leaf pages (DB2 7.1 and later only) |
| DMRAUTOCUST | none | NO | Integer | DMR column used only as a marker during AutoCustomization; all columns following this one in the create table member are not used in the reports (NO in *Used in Report* column) and can be deleted as a group as an AutoCustomization option |

# DMRADxxx—DDF Accounting Detail and Summary Tables

DMRADxxx is an optional table for distributed processing accounting records. Each row in the table represents information about one DDF location for one accounting record.

Table 8. DDF Accounting Record - Detail and Summary

| Performance Reporter Column Name | SMF Field Name | Used in Report | Field Type | Description |
|---|---|---|---|---|
| SYSTEMID | SM101SID | YES | Char(4) | System ID (SMF ID) |
| SUBSYSTEM | SM101SSI | YES | Char(4) | Subsystem ID (DB2 subsystem name) |
| PLANNAME | QWHCPLAN | YES | Char(8) | Plan name |
| AUTHID | QWHCAID | YES | Char(8) | Authorization ID |
| CONNECTION | QWHCCN | YES | Char(8) | Connection name |
| CORRID | QWHCCV | YES | Char(12) | Correlation ID |
| ORIGPRIMID | QWHCOPID | YES | Char(8) | Original primary ID |
| LUWIDNID | QWHSNID | YES | Char(8) | LUWID - Network ID |
| LUWIDLUNM | QWHSLUNM | YES | Char(8) | LUWID - Logical unit name |
| LUWIDINST | QWHSLUUV | YES | Char(6) | LUWID - Instance ID |
| LUWIDCOMIT | QWHSLUCC | YES | Real | LUWID - Commit count |
| CONNTYPE | QWHCATYP | YES | Char(8) | Connecting system type code |
| DATETIME | QWACESC | YES | Timestamp | Date and time record was created |
| DATE | QWACESC | YES | Date | Date record was created |
| YEAR | QWHSSTCK | YES | Char(4) | Year record was created |
| MONTH | QWACESC | YES | Char(2) | Month record was created |
| DAY | QWACESC | YES | Char(2) | Day record was created |
| TIME | QWACESC | YES | Time | Time record was created |
| HOUR | QWACESC | YES | Char(2) | Hour record was created |
| DAYOFWEEK# | QWACESC | NO | Smallint | Relative day of week, 1 to 7, where Monday=1 and Sunday=7 |
| DAYOFWEEK | QWACESC | NO | Char(3) | MON, TUE, WED, THU, FRI, SAT, SUN |
| WEEK# | QWACESC | NO | Integer | Relative week number |
| CPUSUCONV | QWACSUCV | NO | Integer | CPU service unit conversion factor (DB2 5.1 and later only) |
| LOCATION | QWHSLOCN | YES | Char(16) | Local location name (DB2 subsystem ID if not defined) |

Table 8. DDF Accounting Record - Detail and Summary (Continued)

| Performance Reporter Column Name | SMF Field Name | Used in Report | Field Type | Description |
|---|---|---|---|---|
| GROUPNAME | QWHADSGN | YES | Char(8) | Data sharing group name |
| MEMBERNAME | QWHAMEMN | YES | Char(8) | Data sharing member name |
| FIRSTPKG | QPACPKID | YES | Char(18) | First package or DBRM executed (planname if accounting class 7 not active) |
| ACCTTOKN | QWHCTOKN | NO | Char(22) | Accounting token for CICS<br>This field applies to the CICS Attachment Facility, RRSAF, and database access threads. For database access threads, this is the value that is received from the requester system. If the connection to the requester system is through DB2 private protocols, this accounting value is identical to the accounting value used at the requester system. If the connection to the requester system is through DRDA protocols, this accounting value is determined from the first 22 bytes of the correlation token (CRRTKN) value of the access relational database (ACCRDB) command received from the requester system during connect processing. |
| ENDUSERID | QWHCEUID | NO | Char(16) | Optional work station end user ID (DB2 5.1 and later)<br>This can be different from the authorization ID used to connect to DB2. This field contains blanks if the client did not supply this information. |
| ENDUSERTX | QWHCEUTX | NO | Char(32) | Optional end user's transaction or application name that identifies the application that is currently running, not the product that is used to run the application (DB2 5.1 and later)<br>This field contains blanks if the client did not supply this information. |

Table 8. DDF Accounting Record - Detail and Summary (Continued)

| Performance Reporter Column Name | SMF Field Name | Used in Report | Field Type | Description |
|---|---|---|---|---|
| ENDUSERWN | QWHCEUWN | NO | Char(18) | Optional end user's workstation name (DB2 5.1 and later)<br>This field contains blanks if the client did not supply this information. |
| REQLOCATION | QWHDRQNM | NO | Char(8) | Requestor Location |
| REQPROD | QWHDPRID | NO | Char(8) | Distributed transaction requestor product ID |
| REQPRODREL | QWHDPRID | NO | Char(8) | VvvRrrMm - version, release, modification level of the requestor for a distributed transaction |
| REQSERV | QLACFLGS | NO | Char(9) | REQUESTOR or SERVER |
| METHOD | QLACFLGS<br>QWHCATYP | NO | Char(4) | APPL or SYST or BOTH |
| TRACEMASK | QWHSMTN | NO | Integer | Active trace mask |
| PSTNUMBER | QWHCCV | NO | Char(4) | PST number - IMS only |
| PSBNAME | QWHCCV | NO | Char(8) | PSB name - IMS only |
| CICSTRAN | QWHCCV | NO | Char(4) | Transaction code - CICS only |
| CORRNAME | QWHCCV | YES | Char(8) | User ID, Jobname, CICS TRNID, or IMS PSBNAME |
| NETWORKID | QWACNID | YES | Char(16) | Network ID |
| INTERVAL | none | YES | Integer | Interval for the summary accounting table |
| TRANSCNT | none | YES | Integer | Transaction thread count for the summary accounting table |
| IFCIDSEQ# | QWHSISEQ | NO | Integer | IFCID sequence number |
| DDFLOCATION | QLACLOCN | YES | Char(16) | Location name of the remote site |
| SQLSENT | QLACSQLS | YES | Real | SQL statements sent |
| SQLRECV | QLACSQLR | YES | Real | SQL statements received |
| ROWSENT | QLACROWS | YES | Real | Rows sent |
| ROWRECV | QLACROWR | YES | Real | Rows received |
| BYTESENT | QLACBYTS | YES | Real | Bytes sent |
| BYTERECV | QLACBYTR | YES | Real | Bytes received |
| CONVSENT | QLACCNVS | YES | Real | Conversations initiated from this site |

Table 8. DDF Accounting Record - Detail and Summary (Continued)

| Performance Reporter Column Name | SMF Field Name | Used in Report | Field Type | Description |
|---|---|---|---|---|
| CONVRECV | QLACCNVR | YES | Real | Conversations initiated to this site |
| MSGSSENT | QLACMSGS | YES | Real | Messages sent |
| MSGSRECV | QLACMSGR | YES | Real | Messages received |
| TRANSENT | QLACTRNS | YES | Real | Transactions migrated to remote |
| TRANRECV | QLACTRNR | YES | Real | Transactions migrated from remote |
| COMMITSENT | QLACCOMS | YES | Real | Commits sent |
| COMMITRECV | QLACCOMR | YES | Real | Commits received |
| ABORTSENT | QLACABRS | YES | Real | Rollback requests sent |
| ABORTRECV | QLACABRR | YES | Real | Rollback requests received |
| CONVQUED | QLACCNVQ | YES | Real | Conversation requests queued |
| ELAPSWAIT | QLACCPUL | YES | Decimal(15,6) | Elapsed time remote wait |
| ELAPSDATAW | QLACCPUR | YES | Decimal(15,6) | Elapsed wait for remote database |
| CPUDATAR | QLACDBAT | YES | Decimal(15,6) | CPU time in remote database |
| SWLIMBLK | QLACCBLB | NO | Real | Switch to limited block protocol |
| SQLBOUND | QLACRBND | NO | Real | SQL bound for remote access |
| ROWSBUFF | QLACBROW | NO | Real | Number of rows in buffer |
| BLKSENT | QLACBTBF | YES | Real | Blocks sent using block fetch |
| BLKRECV | QLACBRBF | YES | Real | Blocks received using block fetch |
| CONVALLOCS | QLACCNVA | NO | Real | Successful conversation allocations |
| CONVTERMS | QLACCNVT | NO | Real | Successful conversation terminations |
| MAXCONV | QLACCIEL | NO | Integer | Maximum conversations open |
| REQUSTPRSE | QLACPRSE | YES | Real | Number of PREPARE requests sent to participant |
| REQUSTPRRC | QLACPRRC | YES | Real | Number of PREPARE requests received from coordinator |
| REQUSTLASE | QLACLASE | YES | Real | Number of LAST AGENT requests sent to coordinator |
| REQUSTLARC | QLACLARC | YES | Real | Number of LAST AGENT requests received from initiator |
| REQUSTCRSE | QLACCRSE | YES | Real | Number of COMMIT requests sent to participant |

Table 8. DDF Accounting Record - Detail and Summary (Continued)

| Performance Reporter Column Name | SMF Field Name | Used in Report | Field Type | Description |
|---|---|---|---|---|
| REQUSTCRRC | QLACCRRC | YES | Real | Number of COMMIT requests received from coordinator |
| REQUSTBKSE | QLACBKSE | YES | Real | Number of BACKOUT requests sent to participant |
| REQUSTBKRC | QLACBKRC | YES | Real | Number of BACKOUT requests received from coordinator |
| RESPONRRSE | QLACRRSE | YES | Real | Number of FORGET responses sent to coordinator |
| RESPONRRRC | QLACRRRC | YES | Real | Number of FORGET responses received from participant |
| RESPONVYSE | QLACVYSE | YES | Real | Number of request COMMIT responses sent to coordinator |
| RESPONVYRC | QLACVYRC | YES | Real | Number of request COMMIT responses received from participant |
| RESPONVNSE | QLACVNSE | YES | Real | Number of BACKOUT responses sent to coordinator |
| RESPONVNRC | QLACVNRC | YES | Real | Number of BACKOUT responses received from participant |
| THDRMTINDT | QLACINDT | YES | Real | Number of threads indoubt with remote location as coordinator |
| THDRMTCPTR | QLACCPTR | YES | Real | Number of COMMIT operations performed with remote location as coordinator |
| THDRMTRBTR | QLACRBTR | YES | Real | Number of ROLLBACK operations performed with remote location as coordinator |
| THDRMTPRID | QLACPRID | YES | Char(8) | Product ID of remote location |
| ELAPSLOTWT | QLACMDWT | YES | Decimal(15,6) | Total elapsed time spent waiting for an available database access agent slot (DB2 5.1 and later only)<br>This wait occurs when DB2 reaches its maximum number of database access agents, and a DBAT must wait for another DBAT to relinquish its slot. |

Table 8. DDF Accounting Record - Detail and Summary (Continued)

| **Performance Reporter Column Name** | **SMF Field Name** | **Used in Report** | **Field Type** | **Description** |
|---|---|---|---|---|
| DMRACSHARE | none | NO | Integer | DMR column used only as a marker during AutoCustomization to delete data sharing columns, if requested |
| DMRAUTOCUST | none | NO | Integer | DMR column used only as a marker during AutoCustomization; all columns following this one in the create table member are not used in the reports (NO in *Used in Report* column) and can be deleted as a group as an AutoCustomization option |

## DMRAPxxx—Package Accounting Detail and Summary Tables

Package accounting detail records are created from SMF 101 records that were created with accounting class 7/8 active.

**Note:** The columns in Table 9 are shown here in the same sequence as the SMF record sections.

Table 9. DMRAPDTL/DMRAPSUM Columns

| Performance Reporter Column Name | SMF Field Name | Used in Report | Field Type | Description |
|---|---|---|---|---|
| SYSTEMID | SM101SID | YES | Char(4) | SMF system ID |
| SUBSYSTEM | SM101SSI | YES | Char(4) | DB2 subsystem ID |
| PLANNAME | QWHCPLAN | YES | Char(8) | Plan name |
| AUTHID | QWHCAID | YES | Char(8) | Primary authorization ID |
| CONNECTION | QWHCCN | YES | Char(8) | Connection name |
| CORRID | QWHCCV | YES | Char(12) | Correlation ID value |
| ORIGPRIMID | QWHCOPID | YES | Char(8) | Original primary authorization ID |
| LUWIDNID | QWHSNID | YES | Char(8) | LUWID - Network ID |
| LUWIDLUNM | QWHSLUNM | YES | Char(8) | LUWID - Logical unit name |
| LUWIDINST | QWHSLUUV | YES | Char(6) | LUWID - Instance ID |
| LUWIDCOMIT | QWHSLUCC | YES | Real | LUWID - Commit count |
| CONNTYPE | QWHCATYP | YES | Char(8) | Connection type |
| DATETIME | QWACESC | YES | Timestamp | Date and time the record was created |
| DATE | QWACESC | YES | Date | Date record was created |
| YEAR | QWHSSTCK | YES | Char(4) | Year record was created |
| MONTH | QWACESC | YES | Char(2) | Month record was created |
| DAY | QWACESC | YES | Char(2) | Day record was created |
| TIME | QWACESC | YES | Time | Time record was created |
| HOUR | QWACESC | YES | Char(2) | Hour record was created |
| DAYOFWEEK# | QWACESC | NO | Smallint | Relative day of week, 1 to 7, where Monday=1 and Sunday=7 |
| DAYOFWEEK | QWACESC | NO | Char(3) | MON, TUE, WED, THU, FRI, SAT, SUN |
| WEEK# | QWACESC | NO | Integer | Relative week number |
| CPUSUCONV | QWACSUCV | NO | Integer | CPU service unit conversion factor (DB2 5.1 and later only) |

Table 9. DMRAPDTL/DMRAPSUM Columns (Continued)

| Performance Reporter Column Name | SMF Field Name | Used in Report | Field Type | Description |
|---|---|---|---|---|
| LOCATION | QWHSLOCN | YES | Char(16) | Local location name (DB2 subsystem ID if not defined) |
| GROUPNAME | QWHADSGN | YES | Char(8) | Data sharing group name |
| MEMBERNAME | QWHAMEMN | YES | Char(8) | Data sharing member name |
| FIRSTPKG | QPACPKID | YES | Char(18) | First package or DBRM executed (planname if accounting class 7 not active) |
| ACCTTOKN | QWHCTOKN | NO | Char(22) | Accounting token for CICS<br>This field applies to the CICS Attachment Facility, RRSAF, and database access threads. For database access threads, this is the value that is received from the requester system. If the connection to the requester system is through DB2 private protocols, this accounting value is identical to the accounting value used at the requester system. If the connection to the requester system is through DRDA protocols, this accounting value is determined from the first 22 bytes of the correlation token (CRRTKN) value of the access relational database (ACCRDB) command received from the requester system during connect processing. |
| ENDUSERID | QWHCEUID | NO | Char(16) | Optional work station end user ID (DB2 5.1 and later)<br>This can be different from the authorization ID used to connect to DB2. This field contains blanks if the client did not supply this information. |
| ENDUSERTX | QWHCEUTX | NO | Char(32) | Optional end user's transaction or application name that identifies the application that is currently running, not the product that is used to run the application (DB2 5.1 and later)<br>This field contains blanks if the client did not supply this information. |

Table 9. DMRAPDTL/DMRAPSUM Columns (Continued)

| Performance Reporter Column Name | SMF Field Name | Used in Report | Field Type | Description |
|---|---|---|---|---|
| ENDUSERWN | QWHCEUWN | NO | Char(18) | Optional end user's workstation name (DB2 5.1 and later)<br>This field contains blanks if the client did not supply this information. |
| TRACEMASK | QWHSMTN | NO | Integer | Active trace mask |
| PSTNUMBER | QWHCCV | YES | Char(4) | PST number - IMS only |
| PSBNAME | QWHCCV | YES | Char(8) | PSB name - IMS only |
| CICSTRAN | QWHCCV | YES | Char(4) | Transaction code - CICS only |
| CORRNAME | QWHCCV | YES | Char(8) | User ID, Jobname, CICS TRNID, or IMS PSBNAME |
| REQLOCATION | QWHDRQNM | NO | Char(16) | Requestor location name for a distributed transaction |
| REQPROD | QWHDPRID | NO | Char(8) | Distributed transaction requestor product ID |
| REQPRODREL | QWHDPRID | NO | Char(8) | VvvRrrMm - version, release, modification level of the requestor for a distributed transaction |
| NETWORKID | QWACNID | YES | Char(16) | Network ID |
| INTERVAL | none | YES | Integer | Interval used only for the summary accounting package tables |
| TRANSCNT | none | YES | Integer | Transaction thread count for the summary accounting package tables |
| IFCIDSEQ# | QWHSISEQ | NO | Integer | IFCID sequence number |
| OCCURRENCES | QPACRECN | YES | Smallint | Number of occurrences (summary only) |
| PKGFLAG | QPACFLGS | YES | Char(4) | PKG for a package, DBRM for a DBRM |
| EXECLOCATION | QPACLOCN | YES | Char(16) | Remote location name where package was executed (blank if executed locally) |
| COLLECTIONID | QPACCOLN | YES | Char(18) | Package collection ID |
| PROGRAMNAME | QPACPKID | YES | Char(18) | Program name (package ID or DBRM) |
| CONSISTOKEN | QPACCONT | YES | Char(16) | Consistency token (hexadecimal) |
| SQLCOUNT | QPACSQLC | YES | Real | SQL requests count |

Table 9. DMRAPDTL/DMRAPSUM Columns (Continued)

| Performance Reporter Column Name | SMF Field Name | Used in Report | Field Type | Description |
|---|---|---|---|---|
| ELAPSEPKG | QPACSCT | YES | Decimal(15,6) | Total elapsed execution time for this package/DBRM |
| CPUTCBPKG | QPACTJST | YES | Decimal(15,6) | Total TCB CPU time for this package/DBRM |
| ENTEXEVENT | QPACARNA | YES | Real | Number of DB2 entry/exit events |
| EVTSYNCIO | QPACARNE | YES | Real | Number of wait events for synchronous I/O |
| ELAPSYNCIO | QPACAWTI | YES | Decimal(15,6) | Elapsed wait time for synchronous I/O |
| ELPLOCK | QPACAWTL | YES | Decimal(15,6) | Elapsed wait time for lock or latch |
| ELPOTHREAD | QPACAWTR | YES | Decimal(15,6) | Elapsed wait time for other read I/O |
| ELPOTHWRIT | QPACAWTW | YES | Decimal(15,6) | Elapsed wait time for other write I/O |
| ELPUNITSW | QPACAWTE | YES | Decimal(15,6) | Elapsed wait time for unit switch |
| ELPARCQIS | QPACALOG | YES | Decimal(15,6) | Elapsed wait time for archive log quiesce |
| EVTLOCK | QPACARNL | YES | Real | Number of wait events for lock or latch |
| EVTOTHREAD | QPACARNR | YES | Real | Number of wait events for other read I/O |
| EVTOTHWAIT | QPACARNW | YES | Real | Number of wait events for other write I/O |
| EVTUNITSW | QPACARNS | YES | Real | Number of waits for unit switch |
| EVTARCQIS | QPACALCT | YES | Real | Number of waits for archive log quiesce |
| EVTDRAIN | QPACARND | YES | Real | Number of waits for drain locks |
| ELPDRAIN | QPACAWDR | YES | Decimal(15,6) | Elapsed wait time for a drain |
| ELPCLAIM | QPACAWCL | YES | Decimal(15,6) | Elapsed wait time for claim release |
| EVTCLAIM | QPACARNC | YES | Real | Number of wait events for claim release |
| ELPARCREAD | QPACAWAR | YES | Decimal(15,6) | Elapsed wait time for an archive read from tape |
| EVTARCREAD | QPACANAR | YES | Real | Number of wait trace events processed for archive read |
| ELPPGLAT | QPACAWTP | YES | Decimal(15,6) | Elapsed wait time for page latch contention |

Table 9. DMRAPDTL/DMRAPSUM Columns (Continued)

| Performance Reporter Column Name | SMF Field Name | Used in Report | Field Type | Description |
|---|---|---|---|---|
| EVTPGLAT | QPACARNH | YES | Real | Number of wait events for page latch contention |
| GBLMSGELAP | QPACAWTG | YES | Decimal(15,6) | Data sharing elapsed wait time sending messages |
| GBLMSGEVNT | QPACARNG | YES | Real | Data sharing waits sending messages |
| GBLLOKELAP | QPACAWTJ | YES | Decimal(15,6) | Data sharing elapsed wait time global lock contention |
| GBLLOKEVNT | QPACARNJ | YES | Real | Data sharing waits for global lock contention |
| SPWAITELAP | QPACCAST | YES | Decimal(15,6) | Stored procedure elapsed time waiting for TCB (accounting class 8) |
| SPWAITCNT | QPACCANM | YES | Real | Stored procedure waits for a TCB |
| SPROCCNT | QPACSPNS | NO | Real | Number of stored procedures executed (DB2 6.1 and later only)<br>This value is calculated only if accounting class 8 is active. |
| FUNCWAIT | QPACUDST | NO | Decimal(15,6) | Total elapsed time spent waiting for an available TCB before the user-defined function could be scheduled (DB2 6.1 and later only)<br>This value is calculated only if accounting class 8 is active. |
| FUNCCNT | QPACUDNU | NO | Real | Number of user-defined functions scheduled (DB2 6.1 and later only)<br>This value is calculated only if accounting class 8 is active. |
| NESTSCHEMA | QPACASCH | NO | Char(8) | Schema name under which nested activity (nesting of stored procedures, user-defined functions, or triggers) occurs (DB2 6.1 and later only) |
| NESTNAME | QPACAANM | NO | Char(18) | Name of nested activity (DB2 6.1 and later only) |
| NESTTYPE | QPACAAFG | NO | Char(5) | Type of nested activity:<br>SPROC Stored procedure<br>UDF User-defined function<br>TRIG Trigger execution<br>(DB2 6.1 and later only) |

Table 9. DMRAPDTL/DMRAPSUM Columns (Continued)

| Performance Reporter Column Name | SMF Field Name | Used in Report | Field Type | Description |
|---|---|---|---|---|
| WTETAWTK | QPACAWTK | NO | Decimal(15,6) | Wait time for global contention for child L-locks (DB2 7.1 and later only) |
| WTETAWTM | QPACAWTM | NO | Decimal(15,6) | Wait time for global contention for other L-locks (DB2 7.1 and later only) |
| WTETAWTN | QPACAWTN | NO | Decimal(15,6) | Wait time for global contention for pageset/partition P-locks (DB2 7.1 and later only) |
| WTETAWTO | QPACAWTO | NO | Decimal(15,6) | Wait time for global contention for page P-locks (DB2 7.1 and later only) |
| WTETAWTQ | QPACAWTQ | NO | Decimal(15,6) | Wait time for global contention for other P-locks (DB2 7.1 and later only) |
| WTEVARNK | QPACARNK | NO | Real | Number of events with global contention for child L-locks (DB2 7.1 and later only) |
| WTEVARNM | QPACARNM | NO | Real | Number of events with global contention for other L-locks (DB2 7.1 and later only) |
| WTEVARNN | QPACARNN | NO | Real | Number of events with global contention for pageset/partition P-locks (DB2 7.1 and later only) |
| WTEVARNO | QPACARNO | NO | Real | Number of events with global contention for page P-locks (DB2 7.1 and later only) |
| WTEVARNQ | QPACARNQ | NO | Real | Number of events with global contention for other P-locks (DB2 7.1 and later only) |
| DMRACSHARE | none | NO | Integer | DMR column used only as a marker during AutoCustomization to delete data sharing columns, if requested |
| DMRAUTOCUST | none | NO | Integer | DMR column used only as a marker during AutoCustomization; all columns following this one in the create table member are not used in the reports (NO in *Used in Report* column) and can be deleted as a group as an AutoCustomization option |

# Chapter 18. DB2 Audit Tables

This chapter provides detailed information about each of the records in the Performance Reporter performance data tables you can use to produce audit reports.

The audit records are stored in eight tables:

| | |
|---|---|
| DMRAUSUM | Summary |
| DMRAUFAL | Authorization Failures |
| DMRAUGRV | Authorization Control—GRANTs / REVOKEs |
| DMRAUDDL | DDL Access |
| DMRAUDML | DML Access |
| DMRAUDMB | DML at BIND |
| DMRAUCHG | Authorization ID Change |
| DMRAUUTL | Utility Access |

## DMRAUSUM—Audit Summary Table

The DMRAUSUM table is used to produce the Audit Summary and Audit Detail Reports. It requires audit classes 1 through 8 and counts occurrences of IFCIDs 23, 24, 25, 55, 83, 87, 140-145, and 169. It also contains category and type information.

**Note:** The columns in Table 10 are shown in the same sequence as the SMF record sections. All columns not used (NO in *Used in Report* column below) in the predefined reports are defined at the end of the actual DB2 table after the dummy column, DMRAUTOCUST. These columns can be deleted as a group as an AutoCustomization option, or individually before the table is created. Data sharing columns are defined after the dummy column, DMRACSHARE; they can be deleted as a group as an AutoCustomization option as well.

Table 10. DMRAUSUM Columns

| Performance Reporter Column Name | SMF Field Name | Used in Report | Field Type | Description |
|---|---|---|---|---|
| SYSTEMID | SM102SID | YES | Char(4) | System ID (SMF ID) |
| SUBSYSTEM | SM102SSI | YES | Char(4) | Subsystem ID (DB2 subsystem name) |
| PLANNAME | QWHCPLAN | YES | Char(8) | Plan name |
| AUTHID | QWHCAID | YES | Char(8) | Primary authorization ID |
| CONNECTION | QWHCCN | NO | Char(8) | Connection name |
| DATETIME | QWACESC | YES | Timestamp | Date/timestamp - end thread time |
| DATE | QWACESC | YES | Date | Date from datetime |
| YEAR | QWACESC | YES | Char(4) | Year from datetime |
| MONTH | QWACESC | NO | Char(2) | Month from datetime |
| DAY | QWACESC | NO | Char(2) | Day from datetime |
| TIME | QWACESC | YES | Time | Time from datetime |
| HOUR | QWACESC | NO | Char(2) | Hour from datetime |
| DAYOFWEEK# | QWACESC | NO | Smallint | Relative day of week, 1 to 7, where Monday=1 and Sunday=7 |
| DAYOFWEEK | QWACESC | NO | Char(3) | MON, TUE, WED, THU, FRI, SAT, SUN |
| WEEK# | QWACESC | NO | Integer | Relative week number |
| LOCATION | QWHSLOCN | YES | Char(16) | Local location name (DB2 subsystem ID if not defined) |
| GROUPNAME | QWHADSGN | YES | Char(8) | Data sharing group name (DB2 4.1 and later only) |

Table 10. DMRAUSUM Columns (Continued)

| Performance Reporter Column Name | SMF Field Name | Used in Report | Field Type | Description |
|---|---|---|---|---|
| MEMBERNAME | QWHAMEMN | YES | Char(8) | Data sharing member name (DB2 4.1 and later only) |
| ACCTTOKN | QWHCTOKN | NO | Char(22) | Accounting token for CICS |
| PSTNUMBR | QWHCCV | NO | Char(4) | PST number - IMS only |
| PSBNAME | QWHCCV | NO | Char(8) | PSB name - IMS only |
| CICSTRAN | QWHCCV | NO | Char(4) | Transaction code - CICS only |
| CORRID | QWHCCV | NO | Char(12) | Correlation ID |
| ORIGPRIMID | QWHCOPID | NO | Char(8) | Original primary ID |
| CONNTYPE | QWHCATYP | NO | Char(8) | Connection type |
| REQLOCATION | QWHDRQNM | NO | Char(16) | Requestor location name |
| AUTHFAIL | QW0140 | YES | Smallint | Count of authorization failures |
| AUTHCNTL | QW0141 | YES | Smallint | Count of explicit GRANTs / REVOKEs |
| AUTHGRANTS | QW0141AC | NO | Smallint | Count of explicit GRANTs |
| AUTHREVOKE | QW0141AC | NO | Smallint | Count of explicit REVOKEs |
| DDLACCESS | QW0142 | YES | Smallint | Count of CREATEs / DROPs / ALTERs |
| DMLREADS | QW0144 | YES | Smallint | Count of audited first READs |
| DMLWRITES | QW0143 | YES | Smallint | Count of audited first WRITEs |
| DMLATBIND | QW0145 | YES | Smallint | Count of access detected at BIND |
| ISOLATION LEVEL | QW0145IS | NO | Char(1) | Statement isolation level |
| AUTHCHANGE | QW0055 / 83 / 87 / 169 | YES | Smallint | Count of AUTHID changes |
| AUDITUTILITY | QW0023 / 24 / 25 | YES | Smallint | Count of utility trace records (phases) |
| AUDITUTILRUN | QW0025 | YES | Smallint | Count of utilities executed |
| AUDITCATEGORY | QW00140-145 / 55 / 83 / 87 / 169 / 23-25 | YES | Char(13) | Description of audit category |
| AUDITTYPE | QW00140-145 / 55 / 83 / 87 / 169 / 23-25 | YES | Char(16) | Supplementary description of audited authorization |
| LUWIDNID | QWHSNID | NO | Char(8) | LU of Work - Network ID |

Table 10. DMRAUSUM Columns (Continued)

| **Performance Reporter Column Name** | **SMF Field Name** | **Used in Report** | **Field Type** | **Description** |
|---|---|---|---|---|
| LUWIDLUNM | QWHSLUNM | NO | Char(8) | LU of Work - Logical unit name |
| LUWIDINST | QWHSLUUV | NO | Char(6) | LU of Work - Instance value |
| LUWIDCOMIT | QWHSLUCC | NO | Smallint | LU of Work - COMMIT count |
| TRACEMASK | QWHSMTN | NO | Integer | Active trace mask |
| DMRACSHARE | none | NO | Integer | DMR column used only as a marker during AutoCustomization to delete data sharing columns, if requested |
| DMRAUTOCUST | none | NO | Integer | DMR column used only as a marker during AutoCustomization; all columns following this one in the create table member are not used in the reports (NO in *Used in Report* column) and can be deleted as a group as an AutoCustomization option |

## DMRAUFAL—Authorization Failures Table

The DMRAUFAL table is used to produce the Authorization Failures Report. It requires audit class 1 and defines all the fields in IFCID 140.

Table 11. DMRAUFAL Columns

| Performance Reporter Column Name | SMF Field Name | Used in Report | Field Type | Description |
|---|---|---|---|---|
| SYSTEMID | SM102SID | YES | Char(4) | System ID (SMF ID) |
| SUBSYSTEM | SM102SSI | YES | Char(4) | Subsystem ID (DB2 subsystem name) |
| PLANNAME | QWHCPLAN | YES | Char(8) | Plan name |
| AUTHID | QWHCAID | YES | Char(8) | Primary authorization ID |
| CONNECTION | QWHCCN | NO | Char(8) | Connection name |
| DATETIME | QWACESC | YES | Timestamp | Date/timestamp - end thread time |
| DATE | QWACESC | YES | Date | Date from datetime |
| YEAR | QWACESC | YES | Char(4) | Year from datetime |
| MONTH | QWACESC | NO | Char(2) | Month from datetime |
| DAY | QWACESC | NO | Char(2) | Day from datetime |
| TIME | QWACESC | YES | Time | Time from datetime |
| HOUR | QWACESC | NO | Char(2) | Hour from datetime |
| DAYOFWEEK# | QWACESC | NO | Smallint | Relative day of week, 1 to 7, where Monday=1 and Sunday=7 |
| DAYOFWEEK | QWACESC | NO | Char(3) | MON, TUE, WED, THU, FRI, SAT, SUN |
| WEEK# | QWACESC | NO | Integer | Relative week number |
| LOCATION | QWHSLOCN | YES | Char(16) | Local location name (DB2 subsystem ID if not defined) |
| GROUPNAME | QWHADSGN | YES | Char(8) | Data sharing group name (DB2 4.1 and later only) |
| MEMBERNAME | QWHAMEMN | YES | Char(8) | Data sharing member name (DB2 4.1 and later only) |
| ACCTTOKN | QWHCTOKN | NO | Char(22) | Accounting token for CICS |
| PSTNUMBR | QWHCCV | NO | Char(4) | PST number - IMS only |
| PSBNAME | QWHCCV | NO | Char(8) | PSB name - IMS only |
| CICSTRAN | QWHCCV | NO | Char(4) | Transaction code - CICS only |

Table 11. DMRAUFAL Columns (Continued)

| Performance Reporter Column Name | SMF Field Name | Used in Report | Field Type | Description |
|---|---|---|---|---|
| CORRID | QWHCCV | NO | Char(12) | Correlation ID |
| ORIGPRIMID | QWHCOPID | NO | Char(8) | Original primary ID |
| CONNTYPE | QWHCATYP | NO | Char(8) | Connection type |
| REQLOCATION | QWHDRQNM | NO | Char(16) | Requestor location name |
| PRIVILEGECHKD | QW0140PR | YES | Char(16) | Privilege checked |
| OBJECTTYPE | QW0140OB | YES | Char(10) | Object type |
| SOURCEOWNER | QW0140SC | YES | Char(8) | Source object owner |
| SOURCEOBJECT | QW0140SN | YES | Char(18) | Source object name |
| TARGETOWNER | QW0140TC | YES | Char(8) | Target object owner |
| TARGETOBJECT | QW0140TN | YES | Char(18) | Target object name |
| AUTHIDCHKD | QW0140UR | YES | Char(8) | AUTHID checked |
| LUWIDNID | QWHSNID | NO | Char(8) | LU of Work - Network ID |
| LUWIDLUNM | QWHSLUNM | NO | Char(8) | LU of Work - Logical unit name |
| LUWIDINST | QWHSLUUV | NO | Char(6) | LU of Work - Instance value |
| LUWIDCOMIT | QWHSLUCC | NO | Smallint | LU of Work - COMMIT count |
| TRACEMASK | QWHSMTN | NO | Integer | Active trace mask |
| SQLTEXTLEN | QW0140TX | NO | Smallint | SQL text length |
| SQLTEXT | QW0140TX | NO | Varchar(3200) | SQL text |

## DMRAUGRV—Authorization Control Table

The DMRAUGRV table is used to produce the Authorization Control - GRANTs / REVOKEs Report. It requires audit class 2 and defines all the fields in IFCID 141.

Table 12. DMRAUGRV Columns

| Performance Reporter Column Name | SMF Field Name | Used in Report | Field Type | Description |
|---|---|---|---|---|
| SYSTEMID | SM102SID | YES | Char(4) | System ID (SMF ID) |
| SUBSYSTEM | SM102SSI | YES | Char(4) | Subsystem ID (DB2 subsystem name) |
| PLANNAME | QWHCPLAN | YES | Char(8) | Plan name |
| AUTHID | QWHCAID | YES | Char(8) | Primary authorization ID |
| CONNECTION | QWHCCN | NO | Char(8) | Connection name |
| DATETIME | QWACESC | YES | Timestamp | Date/timestamp - end thread time |
| DATE | QWACESC | YES | Date | Date from datetime |
| YEAR | QWACESC | YES | Char(4) | Year from datetime |
| MONTH | QWACESC | NO | Char(2) | Month from datetime |
| DAY | QWACESC | NO | Char(2) | Day from datetime |
| TIME | QWACESC | YES | Time | Time from datetime |
| HOUR | QWACESC | NO | Char(2) | Hour from datetime |
| DAYOFWEEK# | QWACESC | NO | Smallint | Relative day of week, 1 to 7, where Monday=1 and Sunday=7 |
| DAYOFWEEK | QWACESC | NO | Char(3) | MON, TUE, WED, THU, FRI, SAT, SUN |
| WEEK# | QWACESC | NO | Integer | Relative week number |
| LOCATION | QWHSLOCN | YES | Char(16) | Local location name (DB2 subsystem ID if not defined) |
| GROUPNAME | QWHADSGN | YES | Char(8) | Data sharing group name (DB2 4.1 and later only) |
| MEMBERNAME | QWHAMEMN | YES | Char(8) | Data sharing member name (DB2 4.1 and later only) |
| ACCTTOKN | QWHCTOKN | NO | Char(22) | Accounting token for CICS |
| PSTNUMBR | QWHCCV | NO | Char(4) | PST number - IMS only |
| PSBNAME | QWHCCV | NO | Char(8) | PSB name - IMS only |
| CICSTRAN | QWHCCV | NO | Char(4) | Transaction code - CICS only |

Table 12. DMRAUGRV Columns (Continued)

| **Performance Reporter Column Name** | **SMF Field Name** | **Used in Report** | **Field Type** | **Description** |
|---|---|---|---|---|
| CORRID | QWHCCV | NO | Char(12) | Correlation ID |
| ORIGPRIMID | QWHCOPID | NO | Char(8) | Original primary ID |
| CONNTYPE | QWHCATYP | NO | Char(8) | Connection type |
| REQLOCATION | QWHDRQNM | NO | Char(16) | Requestor location name |
| AUTHIDGR | QW0141OR | YES | Char(8) | ID of Grantor or Revoker |
| ACCESSTYPE | QW0141AC | YES | Char(6) | Access type - GRANT / REVOKE |
| OBJECTTYPE | QW0141OB | YES | Char(10) | Object type |
| SQLCODE | QW0141CO | YES | Smallint | SQL error code |
| GRANTREASONCD | QW0141RE | NO | Char(1) | Reason for GRANT - short code |
| GRANTREASON | QW0141RE | YES | Char(8) | Reason for GRANT |
| LUWIDNID | QWHSNID | NO | Char(8) | LU of Work - Network ID |
| LUWIDLUNM | QWHSLUNM | NO | Char(8) | LU of Work - Logical unit name |
| LUWIDINST | QWHSLUUV | NO | Char(6) | LU of Work - Instance value |
| LUWIDCOMIT | QWHSLUCC | NO | Smallint | LU of Work - COMMIT count |
| TRACEMASK | QWHSMTN | NO | Integer | Active trace mask |
| SQLTEXTLEN | QW0141TX | NO | Smallint | SQL text length |
| SQLTEXT | QW0141TX | YES | Varchar(3200) | SQL text |

## DMRAUDDL—DDL Access Table

The DMRAUDDL table is used to produce the DDL Access Report. It requires audit class 3 and defines all the fields in IFCID 142.

Table 13. DMRAUDDL Columns

| Performance Reporter Column Name | SMF Field Name | Used in Report | Field Type | Description |
|---|---|---|---|---|
| SYSTEMID | SM102SID | YES | Char(4) | System ID (SMF ID) |
| SUBSYSTEM | SM102SSI | YES | Char(4) | Subsystem ID (DB2 subsystem name) |
| PLANNAME | QWHCPLAN | YES | Char(8) | Plan name |
| AUTHID | QWHCAID | YES | Char(8) | Primary authorization ID |
| CONNECTION | QWHCCN | NO | Char(8) | Connection name |
| DATETIME | QWACESC | YES | Timestamp | Date/timestamp - end thread time |
| DATE | QWACESC | YES | Date | Date from datetime |
| YEAR | QWACESC | YES | Char(4) | Year from datetime |
| MONTH | QWACESC | NO | Char(2) | Month from datetime |
| DAY | QWACESC | NO | Char(2) | Day from datetime |
| TIME | QWACESC | YES | Time | Time from datetime |
| HOUR | QWACESC | NO | Char(2) | Hour from datetime |
| DAYOFWEEK# | QWACESC | NO | Smallint | Relative day of week, 1 to 7, where Monday=1 and Sunday=7 |
| DAYOFWEEK | QWACESC | NO | Char(3) | MON, TUE, WED, THU, FRI, SAT, SUN |
| WEEK# | QWACESC | NO | Integer | Relative week number |
| LOCATION | QWHSLOCN | YES | Char(16) | Local location name (DB2 subsystem ID if not defined) |
| GROUPNAME | QWHADSGN | YES | Char(8) | Data sharing group name (DB2 4.1 and later only) |
| MEMBERNAME | QWHAMEMN | YES | Char(8) | Data sharing member name (DB2 4.1 and later only) |
| ACCTTOKN | QWHCTOKN | NO | Char(22) | Accounting token for CICS |
| PSTNUMBR | QWHCCV | NO | Char(4) | PST number - IMS only |
| PSBNAME | QWHCCV | NO | Char(8) | PSB name - IMS only |
| CICSTRAN | QWHCCV | NO | Char(4) | Transaction code - CICS only |

Table 13. DMRAUDDL Columns (Continued)

| Performance Reporter Column Name | SMF Field Name | Used in Report | Field Type | Description |
|---|---|---|---|---|
| CORRID | QWHCCV | NO | Char(12) | Correlation ID |
| ORIGPRIMID | QWHCOPID | NO | Char(8) | Original primary ID |
| CONNTYPE | QWHCATYP | NO | Char(8) | Connection type |
| REQLOCATION | QWHDRQNM | NO | Char(16) | Requestor location name |
| DBID | QW0142DB | YES | Smallint | Database ID |
| TABLEID | QW0142OB | YES | Smallint | Table object ID |
| DBNAME | QW0142DB | YES | Char(8) | Database name |
| TABLENAME | QW0142TN | YES | Char(18) | Table name |
| TABLEOWNER | QW0142OW | YES | Char(8) | Table owner |
| TABLECREATOR | QW0142CR | YES | Char(8) | Table creator |
| STMTTYPE | QW0142AC | YES | Char(8) | SQL statement type |
| LUWIDNID | QWHSNID | NO | Char(8) | LU of Work - Network ID |
| LUWIDLUNM | QWHSLUNM | NO | Char(8) | LU of Work - Logical unit name |
| LUWIDINST | QWHSLUUV | NO | Char(6) | LU of Work - Instance value |
| LUWIDCOMIT | QWHSLUCC | NO | Smallint | LU of Work - COMMIT count |
| TRACEMASK | QWHSMTN | NO | Integer | Active trace mask |
| SQLTEXTLEN | QW0142TX | NO | Smallint | SQL text length |
| SQLTEXT | QW0142TX | YES | Varchar(3200) | SQL text |

## DMRAUDML—DML Access Table

The DMRAUDML table is used to produce the DML Access Report. It requires audit classes 4 and 5 and defines all the fields in IFCIDs 143 and 144.

Table 14. DMRAUDML Columns

| Performance Reporter Column Name | SMF Field Name | Used in Report | Field Type | Description |
|---|---|---|---|---|
| SYSTEMID | SM102SID | YES | Char(4) | System ID (SMF ID) |
| SUBSYSTEM | SM102SSI | YES | Char(4) | Subsystem ID (DB2 subsystem name) |
| PLANNAME | QWHCPLAN | YES | Char(8) | Plan name |
| AUTHID | QWHCAID | YES | Char(8) | Primary authorization ID |
| CONNECTION | QWHCCN | NO | Char(8) | Connection name |
| DATETIME | QWACESC | YES | Timestamp | Date/timestamp - end thread time |
| DATE | QWACESC | YES | Date | Date from datetime |
| YEAR | QWACESC | YES | Char(4) | Year from datetime |
| MONTH | QWACESC | NO | Char(2) | Month from datetime |
| DAY | QWACESC | NO | Char(2) | Day from datetime |
| TIME | QWACESC | YES | Time | Time from datetime |
| HOUR | QWACESC | NO | Char(2) | Hour from datetime |
| DAYOFWEEK# | QWACESC | NO | Smallint | Relative day of week, 1 to 7, where Monday=1 and Sunday=7 |
| DAYOFWEEK | QWACESC | NO | Char(3) | MON, TUE, WED, THU, FRI, SAT, SUN |
| WEEK# | QWACESC | NO | Integer | Relative week number |
| LOCATION | QWHSLOCN | YES | Char(16) | Local location name (DB2 subsystem ID if not defined) |
| GROUPNAME | QWHADSGN | YES | Char(8) | Data sharing group name (DB2 4.1 and later only) |
| MEMBERNAME | QWHAMEMN | YES | Char(8) | Data sharing member name (DB2 4.1 and later only) |
| ACCTTOKN | QWHCTOKN | NO | Char(22) | Accounting token for CICS |
| PSTNUMBR | QWHCCV | NO | Char(4) | PST number - IMS only |
| PSBNAME | QWHCCV | NO | Char(8) | PSB name - IMS only |
| CICSTRAN | QWHCCV | NO | Char(4) | Transaction code - CICS only |

Table 14. DMRAUDML Columns (Continued)

| **Performance Reporter Column Name** | **SMF Field Name** | **Used in Report** | **Field Type** | **Description** |
|---|---|---|---|---|
| CORRID | QWHCCV | NO | Char(12) | Correlation ID |
| ORIGPRIMID | QWHCOPID | NO | Char(8) | Original primary ID |
| CONNTYPE | QWHCATYP | NO | Char(8) | Connection type |
| REQLOCATION | QWHDRQNM | NO | Char(16) | Requestor location name |
| DMLTYPE | QW0143 / 144 | YES | Char(10) | DML type - first READ / WRITE |
| DBID | QW0143 / 144DB | YES | Smallint | Internal database ID |
| PSID | QW0143 / 144PS | YES | Smallint | Internal page set object ID |
| TABLEOBID | QW0143 / 144OB | YES | Smallint | Internal table object ID |
| DBNAME | QW0143 / 144DB | YES | Char(8) | Database name |
| PSNAME | QW0143 / 144PS | YES | Char(8) | Page set name |
| URIDLOGRBA | QW0143 / 144UR | YES | Char(6) | Unit of recovery ID (LOG RBA) |
| LUWIDNID | QWHSNID | NO | Char(8) | LU of Work - Network ID |
| LUWIDLUNM | QWHSLUNM | NO | Char(8) | LU of Work - Logical unit name |
| LUWIDINST | QWHSLUUV | NO | Char(6) | LU of Work - Instance value |
| LUWIDCOMIT | QWHSLUCC | NO | Smallint | LU of Work - COMMIT count |
| TRACEMASK | QWHSMTN | NO | Integer | Active trace mask |

## DMRAUDMB—DML at BIND Table

The DMRAUDMB table is used to produce the DML at BIND Report. It requires audit class 6 and defines all the fields in IFCID 145.

**Note:** The columns in Table 15 are shown in the same sequence as the SMF record sections. All columns not used (NO in *Used in Report* column below) in the predefined reports are defined at the end of the actual DB2 table after the dummy column, DMRAUTOCUST. These columns can be deleted as a group as an AutoCustomization option, or individually before the table is created. Data sharing columns are defined after the dummy column, DMRACSHARE; they can be deleted as a group as an AutoCustomization option as well.

Table 15. DMRAUDMB Columns

| Performance Reporter Column Name | SMF Field Name | Used in Report | Field Type | Description |
|---|---|---|---|---|
| SYSTEMID | SM102SID | YES | Char(4) | System ID (SMF ID) |
| SUBSYSTEM | SM102SSI | YES | Char(4) | Subsystem ID (DB2 subsystem name) |
| PLANNAME | QWHCPLAN | YES | Char(8) | Plan name |
| AUTHID | QWHCAID | YES | Char(8) | Primary authorization ID |
| CONNECTION | QWHCCN | NO | Char(8) | Connection name |
| DATETIME | QWACESC | YES | Timestamp | Date/timestamp - end thread time |
| DATE | QWACESC | YES | Date | Date from datetime |
| YEAR | QWACESC | YES | Char(4) | Year from datetime |
| MONTH | QWACESC | NO | Char(2) | Month from datetime |
| DAY | QWACESC | NO | Char(2) | Day from datetime |
| TIME | QWACESC | YES | Time | Time from datetime |
| HOUR | QWACESC | NO | Char(2) | Hour from datetime |
| DAYOFWEEK# | QWACESC | NO | Smallint | Relative day of week, 1 to 7, where Monday=1 and Sunday=7 |
| DAYOFWEEK | QWACESC | NO | Char(3) | MON, TUE, WED, THU, FRI, SAT, SUN |
| WEEK# | QWACESC | NO | Integer | Relative week number |
| LOCATION | QWHSLOCN | YES | Char(16) | Local location name (DB2 subsystem ID if not defined) |
| GROUPNAME | QWHADSGN | YES | Char(8) | Data sharing group name (DB2 4.1 and later only) |
| MEMBERNAME | QWHAMEMN | YES | Char(8) | Data sharing member name (DB2 4.1 and later only) |

Table 15. DMRAUDMB Columns (Continued)

| Performance Reporter Column Name | SMF Field Name | Used in Report | Field Type | Description |
|---|---|---|---|---|
| ACCTTOKN | QWHCTOKN | NO | Char(22) | Accounting token for CICS |
| PSTNUMBR | QWHCCV | NO | Char(4) | PST number - IMS only |
| PSBNAME | QWHCCV | NO | Char(8) | PSB name - IMS only |
| CICSTRAN | QWHCCV | NO | Char(4) | Transaction code - CICS only |
| CORRID | QWHCCV | NO | Char(12) | Correlation ID |
| ORIGPRIMID | QWHCOPID | NO | Char(8) | Original primary ID |
| CONNTYPE | QWHCATYP | NO | Char(8) | Connection type |
| REQLOCATION | QWHDRQNM | NO | Char(16) | Requestor location name |
| BINDLOCATION | QW0145LN | YES | Char(16) | Bind location |
| COLLECTIONID | QW0145PC | YES | Char(18) | Package collection ID |
| PROGRAM | QW0145PN | YES | Char(8) | Program name |
| PRECOMPTIME | QW0145TS | YES | Char(8) | Precompiler timestamp |
| STMTNO | QW0145SN | YES | Smallint | Statement number |
| STMTTYPE | QW0145ST | YES | Char(10) | SQL statement type |
| SQLCODE | QW0145SC | YES | Smallint | SQL return code |
| DBID | QW0145DB | YES | Smallint | Database ID |
| TABLEID | QW0145OB | YES | Smallint | Table object ID |
| ISOLATION LEVEL | QW0145IS | NO | Char(1) | Statement isolation level |
| LUWIDNID | QWHSNID | NO | Char(8) | LU of Work - Network ID |
| LUWIDLUNM | QWHSLUNM | NO | Char(8) | LU of Work - Logical unit name |
| LUWIDINST | QWHSLUUV | NO | Char(6) | LU of Work - Instance value |
| LUWIDCOMIT | QWHSLUCC | NO | Smallint | LU of Work - COMMIT count |
| TRACEMASK | QWHSMTN | NO | Integer | Active trace mask |
| SQLTEXTLEN | QW0145TX | NO | Smallint | SQL text length |
| SQLTEXT | QW0145TX | YES | Varchar(3200) | SQL text |

Table 15. DMRAUDMB Columns (Continued)

| **Performance Reporter Column Name** | **SMF Field Name** | **Used in Report** | **Field Type** | **Description** |
|---|---|---|---|---|
| DMRACSHARE | none | NO | Integer | DMR column used only as a marker during AutoCustomization to delete data sharing columns, if requested |
| DMRAUTOCUST | none | NO | Integer | DMR column used only as a marker during AutoCustomization; all columns following this one in the create table member are not used in the reports (NO in *Used in Report* column) and can be deleted as a group as an AutoCustomization option |

## DMRAUCHG—Authorization ID Change Table

The DMRAUCHG table is used to produce the Authorization ID Change Report. It requires audit class 7 and defines all the fields in IFCIDs 55, 83, 87, and 169.

Table 16. DMRAUCHG Columns

| Performance Reporter Column Name | SMF Field Name | Used in Report | Field Type | Description |
|---|---|---|---|---|
| SYSTEMID | SM102SID | YES | Char(4) | System ID (SMF ID) |
| SUBSYSTEM | SM102SSI | YES | Char(4) | Subsystem ID (DB2 subsystem name) |
| PLANNAME | QWHCPLAN | YES | Char(8) | Plan name |
| AUTHID | QWHCAID | YES | Char(8) | Primary authorization ID |
| CONNECTION | QWHCCN | NO | Char(8) | Connection name |
| DATETIME | QWACESC | YES | Timestamp | Date/timestamp - end thread time |
| DATE | QWACESC | YES | Date | Date from datetime |
| YEAR | QWACESC | YES | Char(4) | Year from datetime |
| MONTH | QWACESC | NO | Char(2) | Month from datetime |
| DAY | QWACESC | NO | Char(2) | Day from datetime |
| TIME | QWACESC | YES | Time | Time from datetime |
| HOUR | QWACESC | NO | Char(2) | Hour from datetime |
| DAYOFWEEK# | QWACESC | NO | Smallint | Relative day of week, 1 to 7, where Monday=1 and Sunday=7 |
| DAYOFWEEK | QWACESC | NO | Char(3) | MON, TUE, WED, THU, FRI, SAT, SUN |
| WEEK# | QWACESC | NO | Integer | Relative week number |
| LOCATION | QWHSLOCN | YES | Char(16) | Local location name (DB2 subsystem ID if not defined) |
| GROUPNAME | QWHADSGN | YES | Char(8) | Data sharing group name (DB2 4.1 and later only) |
| MEMBERNAME | QWHAMEMN | YES | Char(8) | Data sharing member name (DB2 4.1 and later only) |
| ACCTTOKN | QWHCTOKN | NO | Char(22) | Accounting token for CICS |
| PSTNUMBR | QWHCCV | NO | Char(4) | PST number - IMS only |
| PSBNAME | QWHCCV | NO | Char(8) | PSB name - IMS only |
| CICSTRAN | QWHCCV | NO | Char(4) | Transaction code - CICS only |

Table 16. DMRAUCHG Columns (Continued)

| Performance Reporter Column Name | SMF Field Name | Used in Report | Field Type | Description |
|---|---|---|---|---|
| CORRID | QWHCCV | NO | Char(12) | Correlation ID |
| ORIGPRIMID | QWHCOPID | NO | Char(8) | Original primary ID |
| CONNTYPE | QWHCATYP | NO | Char(8) | Connection type |
| REQLOCATION | QWHDRQNM | NO | Char(16) | Requestor location name |
| AUTHIDCHGTYPE | QW0055 / 83 / 87 / 169 | YES | Char(17) | AUTHID change type |
| CURRENTAUTHID | QW0055OI / QW0169AU | YES | Char(8) | AUTHID change - current AUTHID |
| NEWAUTHID | QW0055NI / QW0169NE | YES | Char(8) | AUTHID change - new AUTHID |
| SQLAUTHID | QW0083QD / QW0087QD | YES | Char(8) | SQL AUTHID - identification / signon |
| ORIGPRIMAUTH | QW0083OP / QW0087OP / QWHCOPID | YES | Char(8) | AUTHID change - original ID |
| STMTSTATUS | QW0055ST / QW0083AD / QW0087AD | YES | Char(8) | AUTHID change status |
| SECAUTHCOUNT | QW0083SA / QW0087SA | NO | Smallint | Number of secondary AUTHIDs |
| SECAUTH1 | QW0083SA / QW0087SA | YES | Char(8) | First secondary AUTHID |
| SECAUTH2 | QW0083SA / QW0087SA | YES | Char(8) | Second secondary AUTHID |
| SECAUTH3 | QW0083SA / QW0087SA | YES | Char(8) | Third secondary AUTHID |
| AUTHXLATTYPE | QW0169TY | YES | Char(8) | Distributed AUTHID inbound / outbound connection messages |
| LUWIDNID | QWHSNID | NO | Char(8) | LU of Work - Network ID |
| LUWIDLUNM | QWHSLUNM | NO | Char(8) | LU of Work - Logical unit name |
| LUWIDINST | QWHSLUUV | NO | Char(6) | LU of Work - Instance value |
| LUWIDCOMIT | QWHSLUCC | NO | Smallint | LU of Work - COMMIT count |
| TRACEMASK | QWHSMTN | NO | Integer | Active trace mask |
| AUTHLOCATION | QW0169LO | NO | Char(16) | Distributed AUTHID change responding location |

Table 16. DMRAUCHG Columns (Continued)

| **Performance Reporter Column Name** | **SMF Field Name** | **Used in Report** | **Field Type** | **Description** |
|---|---|---|---|---|
| AUTHLUNAME | QW0169LU | NO | Char(8) | Distributed AUTHID change responding luname |
| SECAUTHLEN | QW0083SL / QW0087SL | NO | Smallint | Length of secondary AUTHID |
| SECAUTHVAR | QW0083SA / QW0087SA | NO | Varchar(100) | String of secondary AUTHID |

## DMRAUUTL—Utility Access Table

The DMRAUUTL table is used to produce the Utility Access Report. It requires audit class 8 and defines all the fields in IFCIDs 23, 24, and 25.

Table 17. DMRAUUTL Columns

| Performance Reporter Column Name | SMF Field Name | Used in Report | Field Type | Description |
|---|---|---|---|---|
| SYSTEMID | SM102SID | YES | Char(4) | System ID (SMF ID) |
| SUBSYSTEM | SM102SSI | YES | Char(4) | Subsystem ID (DB2 subsystem name) |
| PLANNAME | QWHCPLAN | YES | Char(8) | Plan name |
| AUTHID | QWHCAID | YES | Char(8) | Primary authorization ID |
| CONNECTION | QWHCCN | NO | Char(8) | Connection name |
| DATETIME | QWACESC | YES | Timestamp | Date/timestamp - end thread time |
| DATE | QWACESC | YES | Date | Date from datetime |
| YEAR | QWACESC | YES | Char(4) | Year from datetime |
| MONTH | QWACESC | NO | Char(2) | Month from datetime |
| DAY | QWACESC | NO | Char(2) | Day from datetime |
| TIME | QWACESC | YES | Time | Time from datetime |
| HOUR | QWACESC | NO | Char(2) | Hour from datetime |
| DAYOFWEEK# | QWACESC | NO | Smallint | Relative day of week, 1 to 7, where Monday=1 and Sunday=7 |
| DAYOFWEEK | QWACESC | NO | Char(3) | MON, TUE, WED, THU, FRI, SAT, SUN |
| WEEK# | QWACESC | NO | Integer | Relative week number |
| LOCATION | QWHSLOCN | YES | Char(16) | Local location name (DB2 subsystem ID if not defined) |
| GROUPNAME | QWHADSGN | YES | Char(8) | Data sharing group name (DB2 4.1 and later only) |
| MEMBERNAME | QWHAMEMN | YES | Char(8) | Data sharing member name (DB2 4.1 and later only) |
| ACCTTOKN | QWHCTOKN | NO | Char(22) | Accounting token for CICS |
| PSTNUMBR | QWHCCV | NO | Char(4) | PST number - IMS only |
| PSBNAME | QWHCCV | NO | Char(8) | PSB name - IMS only |
| CICSTRAN | QWHCCV | NO | Char(4) | Transaction code - CICS only |

Table 17. DMRAUUTL Columns (Continued)

| Performance Reporter Column Name | SMF Field Name | Used in Report | Field Type | Description |
|---|---|---|---|---|
| CORRID | QWHCCV | NO | Char(12) | Correlation ID |
| ORIGPRIMID | QWHCOPID | NO | Char(8) | Original primary ID |
| CONNTYPE | QWHCATYP | NO | Char(8) | Connection type |
| REQLOCATION | QWHDRQNM | NO | Char(16) | Requestor location name |
| UTILITYID | QW0023 / 24 / 25ID | YES | Char(16) | Utility ID |
| DBID | QW0023 / 24 / 25DB | YES | Smallint | Database DBID |
| PSID | QW0023 / 24 / 25PD | YES | Smallint | Page set PSID |
| UTILITYNAME | QW0023 / 24 / 25NM | YES | Char(8) | Utility name |
| UTILITYPHASE | QW0023 / 24 / 25PH | YES | Char(8) | Utility phase |
| PREVITEMCOUNT | QW0023 / 24DN / 25DN | YES | Integer | Number of items in the previous phase |
| ITEMTYPE | QW0023 / 24 / 25NM-PH | YES | Char(13) | Item type |
| DBNAME | QW0023 / 24 / 25DB | YES | Char(8) | Database name |
| PSNAME | QW0023 / 24 / 25PD | YES | Char(8) | Page set name |
| LUWIDNID | QWHSNID | NO | Char(8) | LU of Work - Network ID |
| LUWIDLUNM | QWHSLUNM | NO | Char(8) | LU of Work - Logical unit name |
| LUWIDINST | QWHSLUUV | NO | Char(6) | LU of Work - Instance value |
| LUWIDCOMIT | QWHSLUCC | NO | Smallint | LU of Work - COMMIT count |
| TRACEMASK | QWHSMTN | NO | Integer | Active trace mask |

# Part 5. Appendixes

This part contains the following appendixes:

# Appendix A. BBSAMP Data Set Members

To help you understand and use your BMC Software product easily, the BBSAMP data set contains members that you can edit for your site’s use. These members contain macros, sample JCL, sample user exit routines, and sample statements for a variety of functions.

Table 18 describes BBSAMP sample batch report members for MAINVIEW for DB2.

Table 18. BBSAMP Data Set Batch Report Members for MAINVIEW for DB2

| **BBSAMP Member Name** | **Description** |
|---|---|
| DPxxxxxx<br>DZPRxxxx | Sample members for Performance Reporter customization |
| FACxxxxx<br>FSAxxxxx<br>FSTxxxxx | QMF forms for Performance Reporter |
| QACxxxxx<br>QSAxxxxx<br>QSTxxxxx | QMF queries for Performance Reporter |

# Appendix B. BBPARM Data Set Members

Table 19 lists sample members in BBPARM that can be used to generate predefined Performance Reporter accounting and statistics reports.

Table 19. BBPARM Data Set Members for Performance Reporter

| BBPARM Member Name | Description |
|---|---|
| SAxxxxxx | SQL to generate summary accounting reports |
| STxxxxxx | SQL to generate statistics reports |
| AUxxxxxx | SQL to generate audit reports |

# Appendix C. SMF Data Collection

This appendix contains a brief explanation of how to collect SMF data. For more information, see the *IBM DATABASE 2 Administration Guide*.

## Starting an Accounting Trace

To collect the accounting data, you must first start an accounting trace using one of these methods:

- Modify the INSTALL parameter on the DB2 Tracing panel (DSNTIPN) to include ACCOUNTING TRACE.

  **Note:** Accounting trace is included in the default parameter specification.

- Issue a DB2 START TRACE command:

  ```
  -START TRACE(ACCTG) CLASS(n)
  ```

## Starting a Statistics Trace

To collect the statistics data, you must first start a statistics trace using one of these methods:

- Modify the INSTALL parameter on the DB2 Tracing panel (DSNTIPN) to include STATISTICS TRACE.

  **Note:** Statistics trace is included in the default parameter specification.

- Issue a DB2 START TRACE command:

  ```
  -START TRACE(STAT) CLASS(n)
  ```

## Starting an Audit Trace

To collect the audit data, you must first start an audit trace using one of these methods:

- Modify the INSTALL parameter on the DB2 Tracing panel (DSNTIPN) to include AUDIT TRACE.

  **Note:** Audit trace is included in the default parameter specification.

- Issue a DB2 START TRACE command:

  ```
  -START TRACE(AUDIT) CLASS(n)
  ```

To audit a table, you must include the AUDIT clause in the CREATE TABLE or ALTER TABLE statement.

To invoke or remove auditing on tables, use these SQL statements:

`CREATE TABLE AUDIT(ALL)` Enables auditing of a table on first READ, first UPDATE, or change within a COMMIT scope, when `-START TRACE(AUDIT) CLASS(4,5)` is active.

`CREATE TABLE AUDIT(NONE)` Disables auditing of a table, even if `-START TRACE(AUDIT) CLASS(4,5)` is active.

`CREATE TABLE(CHANGES)` Enables auditing of a table on first INSERT, UPDATE, or DELETE within a COMMIT scope, when `-START TRACE(AUDIT) CLASS(5)` is active.

# Part 6. Index

This section contains the index.

# Index

# Notes

*100038719*